8 лучших мини-дистрибутивов с. 16

ВАШ БЕСПЛАТНЫЙ DVD

Попурри из Linux

8 легких систем

А также: Hymera Open » Pardus 2009.1 » Mad Skills Motocross » Ответы

LINUX FORMAT

Главное в мире Linux

Апрель 2010 № 4 (130)

Linux на работе

Будь то домашний офис или большая корпорация, Linux может сохранить деньги вашей организации с. 22

32 страницы учебников на любой вкус!

SELinux

Небольшие правки помогут держать систему на замке с. 72

Просто о звуке

Намаялись с PulseAudio? Вы не одиноки — узнайте, как это должно работать на с. 32

«Хоть мы и явились последними, но всё равно захватили лидерство»

Сорен Хансен подстегнул Ubuntu с. 30

Плюс!

» Что такое MicroXwin
» TonidoPlug: сервер-крошка
» Google Go: язык от авторов Unix

4 607130 825037 10004

» Моделирование
Создайте виртуальную модель плотины СШ ГЭС с. 40

» apt-get
Держите систему в чистоте и порядке — из терминала с. 76

» OpenVZ
Другой (и быстрый) способ виртуализации с. 84

Подписной индекс в каталоге агентства «Роспечать» 20882
Подписной индекс в каталоге «Пресса России» 87974
Подписной индекс в каталоге «Почта России» 16572





Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

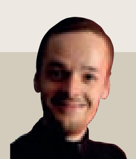
» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

Кто мы

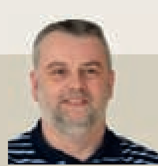
В этом номере мы продвигаем Linux как платформу для офиса. А потому решили спросить: если ваш начальник скептически относится к переходу на свободное ПО, какие аргументы вы используете, чтобы убедить его или ее?



Грэм Моррисон
Просто не слушайте, что говорит босс, и делайте, что считаете нужным. Да, Пол, я уже иду!



Майк Сондерс
Используйте словечки вроде «B2B», «синергия», «проактивность» и «все так делают!»



Нейл Ботвик
Покажите боссу обезьяньи танцы Балмера и спросите: «Вы правда хотите купить ПО у этого человека?»



Эфраин Эрнандес-Мендоса
Раскройте ему глаза распорками и дайте посмотреть концерт Jonas Brothers. Ну, смеха ради.



Эндрю Грегори
Конечно, компактом Perry Como, мягким светом и шампанским. Ну, это если у вас начальница.



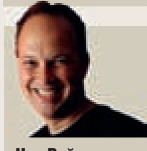
Энди Ченнел
Я сам себе начальство, поэтому мне придется с ним серьезно поговорить. И даже пригрозить увольнением.



Дэвид Картрайт
Устрою забастовку и встану под окнами офиса с плакатом: «Windows – сакс и мастдай». И так далее.



Энди Хадсон
Мой босс разрешил мне Linux, и теперь я на уровне 2: Hurd. Кто устоит перед дважды рекурсивным акронимом?



Ник Вейч
Качайте у него перед глазами маятником, монотонно бубня: «Вы любите Linux, я могу использовать Linux».



Сюзан Линтон
Установите Linux, подберите темку а-ля Редмонд и скажите, что это Windows 8 Pro Turbo Championship.



Шашанк Шарма
Я думаю, здесь можно как-то применить XML. Любую проблему в мире можно решить через XML.



Маянк Шарма
Это не проблема – я перешел на работу, не связанную с компьютерами. Свободная касса!



Мой [рабочий] компьютер

» Готов ли Linux к использованию в офисе? Спросите десять разных людей – и вы получите два разных ответа: ну, а что вы хотели от вопроса, подразумевающего «да» или «нет»? Тем не менее, это означает, что положительный ответ обязательно будет присутствовать. Пусть даже это будет не решительное «да», а «да, но…» – срыв шаблона, начавшийся лет пять-шесть назад и продолжающийся по сей день, налицо.

Почему люди выбирают Linux в качестве ОС для своего бизнеса? Для кого-то (как правило, небольших фирм) это способ сэкономить на лицензиях (а то и просто «пустить пыль в глаза» проверяющим). Но чем более серьезной является организация, тем чаще на первый план выходят такие качества, как надежность и производительность.

Конечно, никто не утверждает, что Linux с Windows «на одно лицо» и переход можно провернуть за вечер так, что наутро никто и не заметит. Это довольно долгий и кропотливый процесс, и те, кто читает наш журнал достаточно давно, могли познакомиться с ним на реальных примерах в рубрике «История успеха». Наши новые подписчики (добро пожаловать!) могут воспользоваться архивом номеров в формате PDF.

Если вы – линуксоид со стажем, то тема этого номера может вызвать недоумение: «Ну и что тут нового?». Но ведь вас, как специалиста, и не надо убеждать в преимуществах перехода на свободное ПО, правильно? Лучше положите журнал на стол начальнику – печатное слово, как известно, имеет особую силу. Ну а вы, тем временем, можете заняться действительно интересными вещами: познакомиться с новым языком Google Go или технологией OpenVZ, например. Да, если в итоге ваш шеф решит перейти на Linux – дайте нам знать.

Валентин Синицын, главный редактор
» info@linuxformat.ru

Как с нами связаться

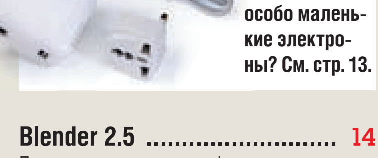
Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru
› Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15
› Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112



Содержание

Весь номер – прямо как на ладони: приятного чтения!

Обзоры



› А что, в этих маленьких компьютерах используются особо маленькие электроны? См. стр. 13.



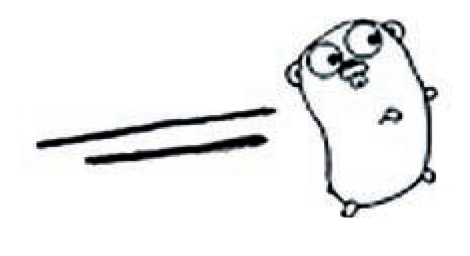
Ваш бизнес должен работать на вас, а не на Microsoft: позвольте Linux помочь вам с. 22

Что за штука...

MicroXwin

Это как X, но компактнее и с добавочными 110 % Win. с. 48

Google Go с. 44



Что поделывает Пайк после того, как изобрел Unix?

Люди говорят



«В облачных вычислениях заинтересованы все, и теперь они свалились на меня.»

Сорен Хансен ощущает груз ответственности с. 30

Попурри дистрибутивов

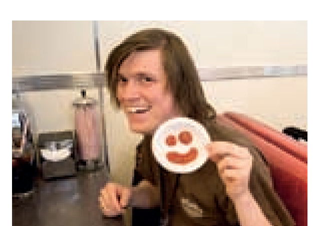
Восемь быстрых Linux'ов, а также

» **Hymera Open** Стильный итальянский Linux
» **Pardus 2009.1** Под эгидой правительства Турции
» **Mad Skills Motocross** Гонки на мотоциклах (демо)
...и множество других приложений и игр! с. 100



Ищите в этом номере...

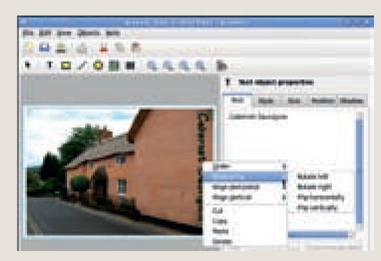
Постоянные рубрики



› А вы не забыли оформить подписку на LXF?

Учебники





› Понюхайте: так пахнет сэкономленное время.

LINUX FORMAT

Новости

ГЛАВНОЕ Войны патентов » Андроиды идут » Ubuntu в облаках » Firefox: в будущее с оптимизмом » Дистрибутивы – не пользователям » Компас-3D для Linux

Продавцы воздуха

Мы давненько не затрагивали вопросы, касающиеся патентов на ПО, а тем временем число новостей на эту тему вышло за все мыслимые рамки.



» Рубрику ведет ЕВГЕНИЙ КРЕСТНИКОВ

Судебные разбирательства в отношении прав на Unix продолжаются. Компания SCO Group получила от группы инвесторов два миллиона долларов, и теперь у нее есть деньги для начала нового судебного процесса против Novell. Слушания стартовали в начале марта; ожидается, что они продлятся три недели. Странный народ американцы: кругом кризис, а они вкладывают кругленькую сумму в «дохлую лошадь», финансируя очередную судебную клоунаду. Видимо, таким нехитрым способом кто-то пытается вставлять палки в колеса компаниям, зарабатывающим на СПО. Кажется, на языке экономистов это называется недобросовестной конкуренцией. Тем временем, Novell удвоила чистую прибыль в первом квартале 2010 финансового года ($20 млн. или 6 центов на акцию против $11 млн. или 3 цента на акцию за аналогичный период 2009 года).

Несмотря на абсурдность самой идеи патентования форматов и алгоритмов, иногда за использование подобных патентов платят. Или меняют их на другие: в недавно опубликованном пресс-релизе Microsoft сообщается, что корпорации удалось договориться о таком обмене с Amazon. Сделка затрагивает используемые в Amazon Linux-серверы, а также патенты, касающиеся электронной читалки Kindle. Думаю, многие помнят эпохальное заявление Microsoft, в котором утверждалось, что СПО нарушает свыше двухсот патентов корпорации. Никакой конкретики миру так и не явили, но под этим соусом Microsoft удалось заключить уже несколько сотен соглашений с крупными компаниями, такими как LG, Samsung, Fuji, Xerox, I-O Data и другими. Почему ведущие игроки рынка соглашаются кормить «тролля»? Не стоит забывать, что многие «жертвы» патентных войн и сами являются держателями патентов. Крупные компании договариваются о взаимном обмене, чтобы избежать судебных исков, а организации, зарабатывающие на СПО, прижимают к ногтю. С другой стороны, некоторые корпорации очень заинтересованы в СПО. Сейчас это напоминает холодную войну между СССР и США, но без четкого деления на два враждующих лагеря: немало крупных держателей патентов активно участвуют и в разработке свободного ПО. Будем надеяться, что патовая ситуация сохранится и далее, а отрасль постепенно перейдет на свободные форматы и алгоритмы.



› Дал МакБрайд [Darl McBride] больше не работает в SCO, но дело его живет.

Web и снова про iPhone

Спецификация HTML 5 еще не готова, и пока не ясно, когда она получит статус рекомендованной. Многих волнует вопрос выбора стандартного кодека для тэга **<video>**. Главные претенденты – проприетарный H.264 и свободный Theora. Первый имеет ряд технических преимуществ, в пользу второго говорит отсутствие лицензионных ограничений и поддержка современными браузерами. Тем временем MPEG-LA продлила возможность бесплатного использования H.264 в потоковом вещании до 2016 года. Изменение лицензионной политики касается только сайтов, раздающих потоковое видео бесплатно. Разработчики прочих проектов (в том числе программных плейеров) с 1 января 2011 года должны будут выплачивать роялти, размер которых пока неизвестен. Ответственность за неуплату распространяется и на конечных пользователей. Фонд свободного ПО (Free Software Foundation) опубликовал открытое письмо, в котором призвал Google открыть технологию VP8 и продвигать ее с помощью сервиса YouTube. Предполагается, что это поможет покончить с монополией H.264.

> «Это напоминает холодную войну между СССР и США.»

Отметилась на данном фронте и компания Apple. Помимо патентных споров с Nokia (затеянных, правда, по инициативе финской стороны), корпорация Джобса обвинила тайваньского производителя HTC в нарушении более 20 принадлежащих ей патентов, использующихся в iPhone. Подробный разбор сути претензий давно сделали сетевые СМИ, так что повторять его здесь полностью мы не будем; но чего стоит лишь один патент, затрагивающий действия с сенсорным экраном для разблокировки устройства! Полагаю, любителям подавать заявки в бюро стоит объяснить, что у человека две руки и десять пальцев, а потому их «изобретение» таковым не является в силу его чисто физиологической очевидности.

Ну, а завершить заметку хочется на мажорной ноте. Истек срок действия патента на технологии работы со шрифтами, используемые в знаменитом патче «byte code interpreter». Да, нам больше не придется пересобирать Freetype! И если раньше исправленным Freetype могли похвастаться лишь некоторые дистрибутивы, есть надежда, что теперь такое положение вещей станет повсеместным.

Весенняя мобилизация

Мобильный Linux, напротив, не может пожаловаться на невнимание с нашей стороны, но тем не менее мы решили осветить некоторые новинки.

Самым крупным событием последних месяцев стала выставка-конференция Mobile World Congress (MWC) 2010, прошедшая в Барселоне во второй половине февраля. Как и ожидалось, на ней было представлено множество устройств (в основном – смартфонов), работающих под управлением ОС Android.

Самым заметным релизом MWC 2010 стал Samsung S8500 Wave – первый телефон, работающий под управлением Bada, новой платформы южнокорейского концерна. Похоже, ребята из Samsung настроены серьезно: о запуске проекта было объявлено в конце 2009 года, а в начале 2010 мы уже видим Bada «в железе».

Корпорация Nokia, напротив, планирует отойти от развития фирменной разработки и объединиться с Intel – вместо Maemo и Moblin появится единая открытая платформа MeeGo, поддерживающая архитектуру Intel и, вероятно, ARM. О немедленном закрытии сольных проектов речи нет; думаю, до появления устройств с MeeGo мы успеем увидеть Maemo 6 (Nokia активно переводит ее на библиотеки *Qt*) и некоторое количество устройств на Moblin. Кстати, на форуме CeBIT Open Source объявили победителей конкурса Linux New Media Award. Платформы Maemo и Moblin заняли второе и третье место в номинации «Лучшее открытое решение для мобильных устройств». Первое взяла платформа Android.

› На MWC 2010 было представлено множество мобильных устройств, работающих под управлением Linux — например, HP Airlife 100.

Конечно, Linux не останется на мобильном рынке в гордом одиночестве – Microsoft анонсировала Windows Phone 7 Series. Корпорация не собирается отказываться и от Windows Mobile 6.5 – на ее основе будет создана Windows Phone Classic, предназначенная для недорогих устройств.

В завершении хотелось бы поговорить о Symbian. Глава Symbian Foundation Ли Уильямс [Lee Williams] в интервью агентству Reuters сообщил, что в этом году Symbian-смартфоны станут дешевле. Что же, это неплохо: в нише бюджетных смартфонов Symbian вне конкуренции – новомодные платформы плохо работают на телефонах за $150.

И все это про него

- Библиотека Qt портируется на платформу Android.
- Adobe представила работающий прототип среды исполнения AIR для Android, предназначенной для запуска web-приложений вне браузера.
- Родерик Коленбрандер [Roderick Colenbrander] перенес на Android *Quake 3*.
- Google выпускает Android Native Development Kit r3.
- Вышел Titanium SDK 1.0.

Adobe входит в LiMo

Платформа LiMo явно уступает Android в смысле популярности у потребителей, но превосходит другие мобильные версии Linux по числу выпущенных моделей, а также поддерживается почти всеми ведущими производителями. В первую очередь Adobe поможет в портировании на LiMo технологии Flash.

Ubuntu в облаках

Сервис Ubuntu One в скором будущем предложит своим пользователям новые функции: что из этого следует?

В предыдущем номере мы уже затронули тему следующего релиза Ubuntu. Тогда мы говорили о технических новшествах, а сейчас самое время перейти к социальным. Операционный директор компании Canonical Джейн Силбер [Jane Silber] заявила, что в 2010 году Ubuntu Linux увеличит свою популярность в качестве настольной и серверной системы. Предполагается, что Ubuntu будет привлекать неофитов «неизбитыми» возможностями рабочей среды, а также оперативностью выпуска обновлений. «С точки зрения работы на потребительском рынке, наши действия будут направлены как на получение определенной рыночной доли у Microsoft, так и на занятие некоторых сегментов, которые в данный момент принадлежат другим дистрибутивам», говорит г-жа Силбер. В значительной степени эти планы связаны с очередной LTS-версией Ubuntu – 10.04, а также с открытием онлайн-сервиса Ubuntu One Music Store, где можно приобрести музыку. Другим важным направлением бизнеса в Canonical считают работу с OEM-партнерами.

Если отбросить словесную мишуру, Canonical делает ставку на Ubuntu 10.04, интерфейс которого будет значительно доработан. Компания также запускает сервис по продаже медиа-контента (в репозиториях Ubuntu Lucid Lynx уже лежит соответствующее дополнение для *Rhytmbox*).

Как и одноименный сервис синхронизации файлов через Интернет, магазин будет основан на облачных технологиях Amazon. Музыка в формате MP3 с достаточно высоким битрейтом (256 kbps) и без цифровой защиты (DRM) предоставлена компанией 7digital – известным поставщиком контента. Объявление о запуске сервиса вызвало бурю негодования в тематических сообществах. В каких только грехах ни обвиняли Canonical, причем безо всяких на то оснований. Сервис Ubuntu One Music Store будет очень либеральным: купленную музыку можно слушать не только на компьютере, но и легально записать на CD или любой портативный плейер с поддержкой MP3. Приобретя песню, скачать ее можно до трех раз (на случай потери данных); кроме того, магазин будет интегрирован с сервисом Ubuntu One, и пользователь сможет хранить записи в своем персональном облаке. Что касается выбора формата – здесь Canonical ограничена условиями партнера, однако Ubuntu Wiki сообщает, что попытки улучшить качество музыки предпринимаются постоянно. В любом случае, если пользователя не устраивают условия работы с магазином – никто не заставляет его покупать там музыку.



› K Store, Sync и Share вскоре, наверное, добавится и Listen: ничего страшного в этом нет.

Вставьте шплинт А в гнездо Б...

Вопреки подозрениям некоторых читателей, мы в курсе, что кроме Ubuntu в мире есть и другие дистрибутивы. Например...

Так уж сложилось, что в новостной колонке мы освещаем только пользовательские дистрибутивы. Пожалуй, время от времени традиции стоит нарушать, и сегодня мы поговорим о реализациях Linux, выделяющихся из общего потока.

В начале марта вышел Linux From Scratch 6.6 – новая версия самого необычного из всех дистрибутивов. Здесь есть архивы с исходными текстами, которые включают определенную версию ядра GNU/Linux, *glibc*, *GCC* и других важных компонентов системы. Есть и LiveCD для сборки и установки (впрочем, в качестве среды для сборки системы можно использовать другие дистрибутивы). Не хватает в LFS только одного – инсталлятора, который заменяет книга Linux From Scratch Book. В общем, тут все как в замечательном рассказе Айзека Азимова, цитатой из которого мы и начали заметку. Конечно, использовать LFS в реальной работе не очень удобно (хотя мы знаем и таких уникумов), но как учебное пособие он идеален.

Следующий наш герой ближе к людям – у дистрибутива Gentoo немало поклонников. Главная его «фишка» – продвинутая система автоматизированной сборки программ из исходных текстов, известная под именем *Portage*. На Gentoo основано несколько интересных дистрибутивов – скажем, вышедший недавно Calculate Linux 10.2. Он полностью совместим с прародителем, но отличается от него многочисленными предварительно сконфигурированными функциями, предназначенными для быстрого развертывания. Кстати, в рамках проекта Calculate Linux запущен новый сервис обзора «портежей» Gentoo. В январе появился еще один подобный сервис – znurt.org; видимо, старый добрый packages.gentoo.org многих уже не устраивает. Напоследок стоит упомянуть выпущенный в конце февраля SystemRescueCD 1.4.0 – LiveCD для восстановления после сбоев, также основанный на Gentoo.

Последняя новинка, о которой я хочу рассказать – Frugalware 1.2 (Locris), дистрибутив, предназначенный для опытных пользователей. Он поддерживает архитектуры i686 и x86_64 (amd64). В качестве менеджера пакетов применяется *pacman* из Arch Linux. Если верить разработчиком, они стремятся к тому, чтобы сделать Frugalware как можно проще, не забывая при этом об удобстве пользователя.

Новости короткой строкой

» Компания Wine@Eteersoft объявила о выпуске пакета Wine@Etersoft CAD 1.1, предназначенного для запуска CAD-систем. Пока обеспечивается работа КОМПАС-3D V10, но в течение года предполагается реализовать поддержку AutoCAD и BricsCAD.

» Открылся магазин атрибутики store.linux.com. Полученные средства пойдут на развитие GNU/Linux.

» NVIDIA отзывает драйверы 196.75.XX, 195.36.08 и 195.36.03 из-за проблем со скоростью вращения вентилятора.

» Саймон Фиппс [Simon Phipps], руководитель направления Open Source компании Sun Microsystems, написал заявление об увольнении.

» Появился релиз *GRUB 1.98*.

Firefox: Пациент скорее жив, чем мертв

Удастся ли *Google Chrome* в одночасье свергнуть *Mozilla Firefox* с пьедестала почета? Мы сомневаемся.

Злые языки утверждают, что золотые дни *Firefox* уже прошли, и его популярность идет на спад. Аналитики связывают этот факт с растущей рыночной долей *Google Chrome*. Лично я никогда не доверял подобным спекуляциям – любую статистику можно подогнать под нужный результат. И хотя пользователи, перешедшие с *Firefox* на *Chrome/Cromium*, мне знакомы (я и сам отношусь к их числу), слухи о «смерти» *Firefox* кажутся несколько преувеличенными. Если судить по обилию новостей от Mozilla – проект живее всех живых. Да и браузером по умолчанию в популярных дистрибутивах *Chromium* еще не стал.

Пока аналитики хоронят проект, в Mozilla вовсю занимаются улучшением поддержки JavaScript в *Firefox*. В конце февраля появилась информация о запуске проекта JägerMonkey, цель которого – интеграция контекстно-ориентированного многопоточного JIT-компилятора в SpiderMonkey, JS-движок браузера *Firefox*. Формирование машинного кода на лету в JägerMonkey осуществляется с помощью открытого ассемблера Nitro JavaScript JIT, разработанного в Apple для WebKit. Если добавить сюда более ранние новости (изменение цикла разработки и новый механизм работы с расширениями), становится ясно, что в Mozilla вовсю готовятся к выпуску четвертой версии *Firefox*, который должен состоятся в конце года. Процесс обновления лицензии MPL (Mozilla Public License), запущенный недавно в Mozilla Foundation, тоже можно считать подготовкой к крупному релизу – обновление закончат в октябре/ноябре 2010 года. В общем, работа кипит, а новые версии выходят по графику. И даже если популярность *Firefox* отчасти снизилась (что спорно), он все равно остается открытым браузером № 1. Что будет дальше – покажет время.



› ***Firefox 4* обещает радикальный пересмотр пользовательского интерфейса: видимо, опыт KDE 4 пугает не всех.**



Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
Его слабости — mass storage, разметка диска и файловые системы.

Неттопия на практике

К теме, нужны ли нам большие машины, я уже обращался в LXF120 – тогда чисто теоретически. Пришло время проверить свои соображения на практике – путём приобретения неттопа в следующем виде: платформа Pegatron со «впрессованным» процессором Atom 330, чипсетом ION (то есть с интегрированным видео уровня GeForce 9xxx), 1 ГБ впаянного ОЗУ и разъёмом SO DIMM с потенциалом ещё на 2 ГБ. Каковой и был реализован – вкупе с обычным ноутбучным винчестером о 5400 об/мин.

В итоге получилась очень милая машинка размером чуть больше двух пачек сигарет, абсолютно бесшумная и холодная, способная функционировать в режиме 24/7 без малейшего вреда для нервов хозяина. Производительность? Рекордов в деле компиляции ядра или тотальной пересборки Gentoo от неё не ждите. Но для большинства пользовательских задач – текстовый редактор и даже процессор типа *OOo*, сёрфинг по Сети и так далее – более чем достаточна.

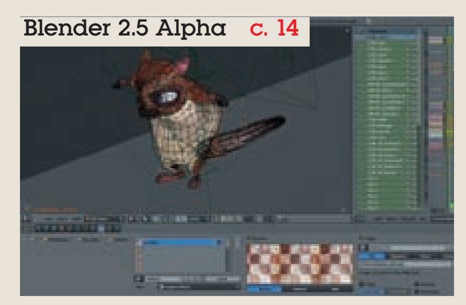
А при чём здесь Linux? Да в общем-то ни при чём; разве что Linux прекрасно на ней работает. В варианте Fedora – с некоторыми ручными пассами при установке фирменных драйверов NVIDIA. Впрочем, преодолимыми и хорошо документированными (скажем, на форуме russianfedora.ru). Ну, а в варианте от Ubuntu – просто из коробки.

В общем, если вам надоел рёв турбин от бессчётного числа кулеров, если вы не занимаетесь непрерывным рендерингом трёхмерных сцен или постоянной пересборкой ядра и мира, не играете в крутые современные игры – может, стоит задуматься о таком варианте? Мне он понравился.

alv@posix.ru

Сегодня мы рассматриваем:

Blender 2.5 Alpha с. 14



› **Так выглядит новый интерфейс Blender; но главные отличия, как обычно, внутри.**

TonidoPlug с. 13



› **Упихайте Linux-сервер в коробку, куда не поместится даже пара ботинок.**

Наш вердикт: Пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибалльной шкале (0 – низшая оценка, 10 – высшая). Как правило, учитываются функциональность, производительность, простота использования и цена, а для бесплатных программ – еще и документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.



Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но если разработчики рекомендуют *Autopackage*, мы следуем этому совету.



Google Earth

Разработчики: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Оправданность цены	9/10

» Если весь мир — сцена, то Google Earth — театр. Простая в использовании, захватывающая и ободряющие практичная программа.

Рейтинг 9/10

Digikam 1.0



Может ли бесплатное ПО тягаться с разорительно дорогими продуктами Adobe и Apple? **Энди Ченнел** добыл супернавороченный фотоаппарат и проверил.

Вкратце

» **Менеджер фотогалерей для KDE. См. также:** *F-Spot*, *Picasa*, *Kalbum*

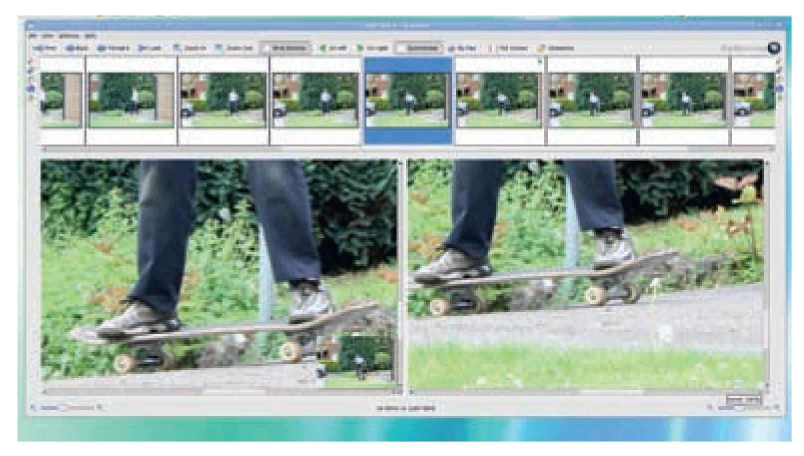
Есть три вида ПО для работы с фотографиями, и все они представлены в Linux. Один полюс – *Picasa*, *iPhoto*, *F-Spot* и им подобные, рассчитанные в основном на управление обширными коллекциями снимков при минимуме редактирования. Другой – пакеты типа *Photoshop* и *GIMP*, обеспечивающие результаты профессионального уровня, но только для единичных фотографий. Между ними расположен широкий спектр ПО для управления крупными фотогалереями и обработки (не обязательно редактирования) изображений. Эти приложения позволяют просматривать фотографии, править цветовые и тональные диапазоны и применять эффекты, но лишены таких функций, как кисти, слои и текстовые инструменты. Принцип работы вкратце таков: создаётся 60 вариантов одной фотографии, из которых выбирается лучший (возможно, после некоторой «рихтовки»).

> «Эффекты применяются очень быстро, даже к большим фото.»

Если вы ищете программу такого типа, обратите внимание на *Digikam*. Она создана для KDE, но неплохо работает и в Gnome. Внешность приложения KDE легко заменить на обличье в стиле Adobe с помощью меню графических тем. Полноэкранный режим позволяет вообще избавиться от интерфейса и целиком сосредоточиться на изображении.

В режиме галереи программа действует элегантно и быстро, позволяя сортировать снимки по датам, камерам, разрешению и другим метаданным. Можно создавать и собственные тэги.

› **Функция Fuzzy Search [Нечеткий поиск] позволяет искать сходные фотографии по цвету, силуэту или эскизу.**



› **Световой стол — отличный способ сравнить RAW-изображение с его JPEG-вариантом. При просмотре можно синхронизировать их масштаб и положение на экране.**

Режим редактирования изображения включается двойным щелчком на фотографии. Предлагается неплохой набор эффектов: от Blur [Размывание] и Sharpen [Резкость] до Spread [Размах], Solarise [Соляризация] и Paint [Краска]. Привыкших к изобилию настроек эффектов в *GIMP* аскетизм опций *Digikam*, пожалуй, разочарует. Но у этих программ разные ниши.

Сравним контрасты

Пусть «вытягивание» снимков и не самое веселое занятие – инструменты *Digikam* работают неплохо, и эффекты применяются очень быстро, даже к большим фото.

Ключевая функция новой версии – Light Table [Световой стол], превосходный инструмент для сравнения фотографий, весьма полезный, когда из сотен снимков требуется отобрать лишь несколько.

Как и полагается продукту, претендующему на звание «профессионального», *Digikam* отлично ладит с RAW-форматами. Он легко справился с несколькими подформатами, которыми «угостили» его наши фотоаппараты.

Программа позволяет контролировать процесс вывода изображений для печати или для публикации в Интернете, но, опять же, хотелось бы иметь больше власти над цветовым балансом, насыщенностью и тональностью.

Для ваших доведенных до совершенства фото предоставляется множество вариантов вывода, в том числе отправка прямо на Facebook, Flickr и другие онлайн-службы, вывод на iPod и создание HTML-галерей для размещения на сайтах. Не помешало бы подобное разнообразие и в сфере импорта – например, функция распознавания лиц (как в *Picasa*) для автоматического назначения тэгов. Но уж те возможности, что есть, работают безупречно.

Для приложения такого типа жизненно важна производительность, ведь приходится обрабатывать множество фотографий с высоким разрешением. *Digikam* шустро перебирает миниатюры, и на нашем Core 2 Duo с 2 ГБ ОЗУ открывал фотографии с приемлемой задержкой. Итак, *Digikam* – шаг вперед относительно *F-Spot*, но то, что вы приобретете в функциональности – потеряете в простоте. LXF



Digikam 1.0

Разработчик: KDE/Digikam
Сайт: www.digikam.org
Цена: Бесплатно на условиях GNU GPL

Функциональность	7/10
Производительность	6/10
Простота использования	7/10
Документация	5/10

» Новая версия программы замахнулась на обработку, но инструментарий мог бы быть богаче.



Рейтинг 6/10

Mandriva InstantOn

Глядя на облегченную версию Mandriva, **Боб Мосс** задумался: вправду ли мал золотник, да дорог – или... просто мал?

Вкратце

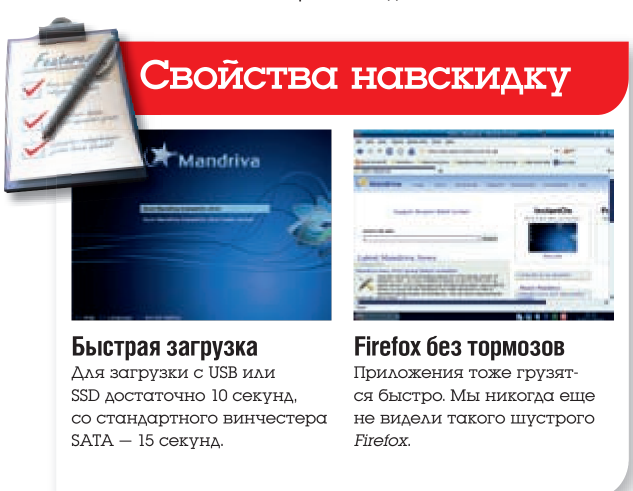
» **Дистрибутив «мгновенной готовности», рассчитанный на двойную загрузку с другой ОС. См. также: Splashtop.**

Все мы озабочены ускорением загрузки – отсюда и разговоры о ПО «мгновенной готовности» (instant-on). Есть теория, что подключать полноценный рабочий стол не обязательно, если всего-то и надо, что проверить почту, прослушать MP3 или выйти в Интернет.

Тут и появляется Mandriva InstantOn... Облегченная версия грузится со скоростью, которая не снилась обычной ОС. Старт с USB или SSD отнимает менее 10 секунд, а с SATA и IDE – около 15. В отличие от других подобных решений, здесь не нужно ничего настраивать – все работает прямо из коробки.

И что же там работает?.. Едва ли не все приложения Mandriva InstantOn можно пересчитать по пальцам одной руки. Непременный *Mozilla Firefox* отвечает за путешествия по Интернету, *Pidgin* и *Skype* обеспечивают обмен сообщениями и голосовую связь, *Mozilla Thunderbird* работает с почтой, *Rhythmbox* играет музыку. Есть еще стандартный медиа-плейер Gnome на основе *Totem* с дополнительными кодеками DivX. Вот так в Mandriva оценили ваши потребности. Очевидных методов расширить предлагаемый список не просматривается, хотя для повседневной работы среднего пользователя этого набора вполне достаточно.

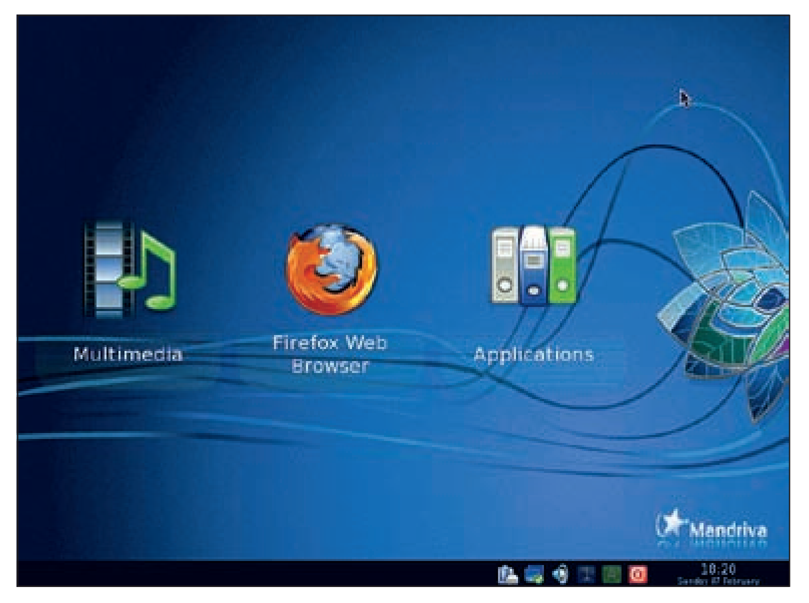
> «Приложения можно пересчитать по пальцам одной руки.»



Свойства навскидку

Быстрая загрузка
Для загрузки с USB или SSD достаточно 10 секунд, со стандартного винчестера SATA — 15 секунд.

Firefox без тормозов
Приложения тоже грузятся быстро. Мы никогда еще не видели такого шустрого *Firefox*.



› **Если вы подбираете интерфейс для нетбука, то проще и удобнее Mandriva InstantOn, пожалуй, не найти.**

Оно нам надо?

Новую версию Mandriva планирует продавать. Тех, кто вносит деньги на поддержку компании, это вряд ли смутит – а вот другим пользователям может не понравиться. Mandriva InstantOn – неплохой дистрибутив, но его соперник, Splashtop, предлагает ту же функциональность бесплатно, да еще при поддержке таких китов, как Acer, Sony и Toshiba (список не полон). А проекты вроде Ubuntu Netbook Remix и Moblin демонстрируют примерно такую же скорость загрузки, обладая большей гибкостью в выборе приложений и кодеков. Все эти продукты также распространяются совершенно бесплатно, и чтобы оправдать свой ценник, InstantOn должен серьезно оторваться от конкурентов.

Удалось ли ему это? Нельзя отрицать, что дистрибутив интуитивен, отличается быстротой и делает все, чего следует ожидать от «продукта быстрого приготовления». Однако в нем не предусмотрено простых средств настройки на свой вкус или инструментов отладки на случай сбоя. Минималистский интерфейс – это, конечно, преимущество, но пользователи с определенными пристрастиями или потребностями в области ПО найдут функциональность InstantOn весьма ограниченной.

Полные Linux-новички, возможно, соблазнятся простотой продукта – ведь он заработает у них с ходу, без лишних телодвижений. Однако нечто подобное у вас получится самим смастерить на Linux-нетбуке, повозившись какую-нибудь пару часов.

В то, чтобы сделать InstantOn максимально легким и быстрым в загрузке, разработчиками явно вложен немалый труд, но коль скоро речь заходит о свободном ПО, выставляемом на продажу, «меньше» отнюдь не всегда бывает «лучше». LXF



LINUX FORMAT Вердикт

Mandriva InstantOn 2010

Разработчик: Mandriva
Сайт: www2.mandriva.com/instanton
Цена: €9,90

Функциональность	6/10
Производительность	9/10
Простота использования	9/10
Документация	5/10

» Идеальный «продукт быстрого приготовления» для фанатов Mandriva, но другие пользователи вряд ли сочтут цену оправданной.

Рейтинг 6/10

Pardus 2009.1



Государственный Linux-дистрибутив? Да – и очень приличный, несмотря на нехватку англоязычной документации, считает **Ник Вейч**.

Вкратце

» Государственный дистрибутив Турции, ни больше ни меньше. Напоминает OpenSUSE и Mandriva.

Ключевое ПО

- » Ядро 2.6.31.11
- » KDE 4.3.4
- » Gnome 2.26.0
- » Firefox 3.5.7
- » GIMP 2.6.8
- » X.org 1.6.5

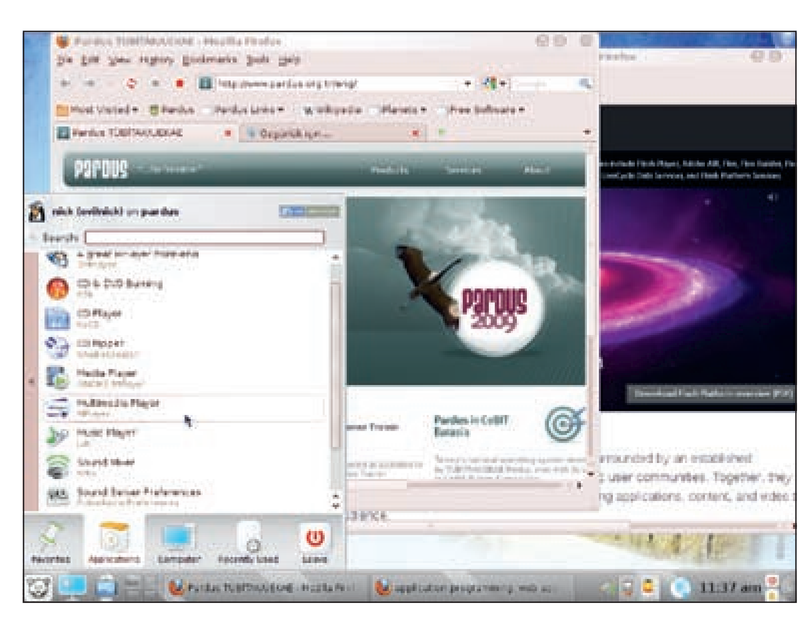
Оцените размах: это Linux-дистрибутив, официально одобренный правительством Турции. Проект основан и поддерживается Советом по научным и технологическим исследованиям Турции с целью создания стандартизованной ОС-платформы для развития и разработки разноплановых приложений. Pardus претендует на независимость, но внешним видом и поведением больше напоминает Mandriva, чем какой-либо другой дистрибутив.

Установка проста. Дистрибутив уместился на одном CD, который пригоден и для работы в Live-режиме, и для установки на винчестер (выбор варианта – в загрузочном меню). Особенно впечатляет настройка после инсталляции: графическая утилита под названием *Kaptan* берет на себя доводку вашего рабочего стола и различных аспектов сетевого окружения.

Малые размеры заставляют усомниться в богатстве набора приложений. Однако после установки вас ждет приятный сюрприз: скажем, медиа-плейеры и ПО для графики не просто представлены, но еще и в достойном выборе (мы нашли *Amarok*, *Juk*, *MPlayer* и *SMPlayer*).

> «Pardus мигом грузится и очень быстро работает с KDE.»

Интересно, что набор пакетов включает драйверы Nvidia, ATI и некоторые медиа-кодеки, то есть к работе можно приступать сразу после установки. Запустив браузер, вы заметите, что он укомплектован *Shockwave*, *Flash*, DivX и другими модулями. Кроме специальных приложений, Pardus использует для настройки *KDE Control Center*, что вовсе не плохо.



» На на одном-единственном CD разместилась масса полезных приложений: новейший *GIMP*, драйверы для видеокарт, ПО для аудио и медиа-кодеки.

Дистрибутив оснащен собственной системой управления пакетами, подобной таковой у Gentoo. Инструмент для сборки и установки пакетов *Pisi* (по-турецки – котенок) удобен в использовании. Написанные на Python инструменты настолько просты, что с их помощью можно без труда портировать в данный формат сотни (а если постараться, то и тысячи) дополнительных пакетов. Существует система для преобразования *RPM*-пакетов в формат *Pisi* – но компилировать самостоятельно будет, вероятно, проще.

Рай для программера

Pardus рассматривается как платформа для разработки и распространения ПО, поэтому изобилует инструментами для программистов. Трудно назвать язык или среду разработки, которые здесь не поддерживаются.

Хотя дистрибутив турецкий, язык по умолчанию – американский английский, а для специальных Pardus-приложений поддерживается еще десяток языков. Одно слабое место у дистрибутива все-таки имеется: это поддержка и документация, которые, по понятным причинам, тяготеют к государственному языку. Главный сайт технической поддержки и сообщества говорят исключительно по-турецки (правда, на неофициальной wiki-странице можно найти несколько руководств и методы устранения самых распространенных проблем).

Стоит упомянуть также о том, что Pardus не только мигом грузится – он очень быстро работает с KDE. Достоверно измерить скорость работы трудно – просто попробуйте KDE 4.x в действии. LXF

Свойства навскидку



Менеджер пакетов Pisi
Удобен в работе, но ограничен репозиторием Pardus (около 1300 пакетов).



Простая настройка
Kaptan – дружелюбная и удобная система настройки рабочего стола.



Вердикт

Pardus 2009.1

Разработчик: TUBITAK – UEKAE
Сайт: www.pardus.org.tr
Цена: Бесплатно на условиях нескольких лицензий

Функциональность	7/10
Производительность	7/10
Простота использования	8/10
Документация	5/10

» Превосходная демонстрация возможностей KDE 4.x, с изобилием готовых функций и мощным арсеналом программиста.

Рейтинг 7/10

Hymera Open



Может ли региональный дистрибутив претендовать на международную значимость? Разбирается **Семен Есилевский**.

Вкратце

» **Дружественный дистрибутив для домашних и корпоративных пользователей. См. также: Mandriva, Linux Mint.**

Коммерческие дистрибутивы занимают в мире Linux особое место. Такие гиганты, как Red Hat и Novell, имеют свою нишу на корпоративном рынке, а для «домашних» пользователей предлагают бесплатные версии своих продуктов. Понятно, что появление нового игрока в коммерческом сегменте не может не вызывать интереса. Итак, встречайте амбициозного новичка – Hymera Linux.

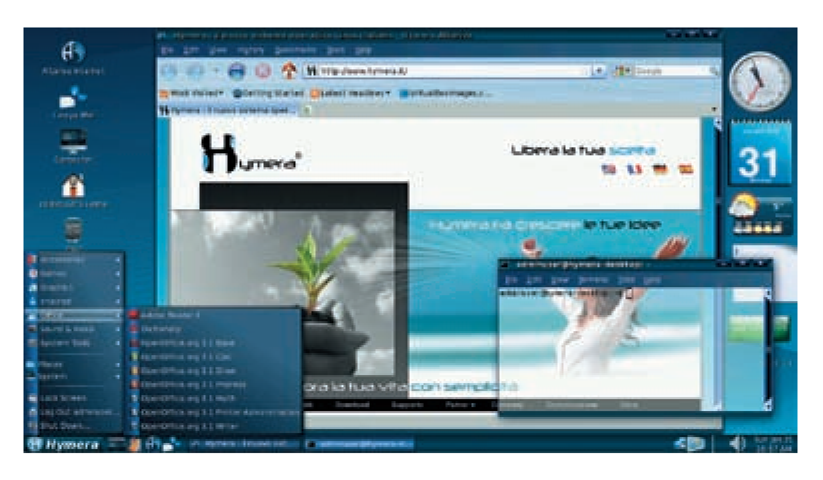
Hymera – молодой итальянский дистрибутив (первый релиз состоялся в 2009 году), призванный развеять миф о сложности Linux для пользователя. Он базируется на Debian, но ядро и многие пакеты серьезно модифицированы. На сегодня доступна только 32-битная версия с рабочей средой Gnome. Из уникальных особенностей Hymera можно отметить утилиту для управления драйверами, возможность легкого «отката» к первоначальным настройкам и два варианта рабочего стола. Существует четыре редакции дистрибутива: бесплатная Hymera Open, ориентированная на «домашних» пользователей Hymera Desktop, Hymera Evolution (LiveUSB) и сервер Hymera Server, который должен появиться в ближайшие месяцы. Платные версии практически не отличаются от бесплатной, но включают техническую поддержку в течение года с гарантией решения любых проблем в течении двух дней.

> «Hymera развеивает миф о сложности Linux для пользователя.»

Несмотря на итальянское происхождение, Hymera не является сугубо «региональным» продуктом – имеются итальянская и английская версии. Язык системы можно поменять на русский после установки, но об официальной поддержке говорить пока не приходится.

Тест-драйв

Англоязычную версию Hymera Open можно скачать с официального сайта в виде ISO-образа. Диск не является LiveCD, а сразу же предлагает установить систему. Процесс установки стандартный, но инсталлятор сделан очень красиво и профессионально, а все шаги детально объяснены.



› **Даже те, для кого красота — не функция, согласятся: Hymera выглядит стильно.**

По скорости загрузки Hymera не рекордсмен, но его не попрекнешь медлительностью. Оригинальный экран загрузки системы выдержан в общей цветовой гамме.

Рабочий стол тоже очень красив. Стандартная компоновка призвана облегчить миграцию для пользователей Windows. Используется единственная панель внизу, с крупной кнопкой меню, а вдоль правого края экрана расположены безделушки-скринлеты. В альтернативной компоновке панель расположена вверху и выглядит более привычно для пользователей Gnome, а внизу находится красивый анимированный док «а-ля Mac OS X». Используется также более «спокойная» тема. Переключение между компоновками – одним щелчком мыши. 3D-эффекты рабочего стола включаются автоматически при наличии мощной видеокарты. Hymera не гнушается проприетарных компонентов: здесь есть драйверы для видео- и беспроводных сетевых карт, *Adobe Reader*, *Skype*, Flash, мультимедиа-кодеки и Java. В соответствии с требованиями по использованию товарных знаков Mozilla, в Hymera выполнили ребрендинг браузера *Firefox* и почтового клиента *Thunderbird*, обеспечив их идеальную интеграцию с системной темой.

За установку и удаление программ отвечает привычный и функциональный *Synaptic*. Для настройки системы используются как стандартные для Gnome, так и оригинальные утилиты (например, для работы с драйверами и модулями ядра, сбора информации о системе, настройки беспроводного доступа).

Итог

Hymera работает «из коробки», практически не требуя настройки. Его прямыми конкурентами являются, например, Linux Mint и Mandriva. Революционных инноваций в Hymera нет, но стандартные функции тут реализованы на «отлично». Дистрибутив имеет яркую индивидуальность за счет прекрасного визуального оформления и пристального внимания к мелочам. Платные варианты для русскоязычных пользователей не оправданы из-за высокой цены и отсутствия поддержки на родном языке, однако бесплатный вариант Hymera Open – выбор достойный. Красота и элегантность интерфейса, предустановленные «закрытые» драйвера, кодеки, Java и Flash, а также широкий выбор программ порадуют и новичков, и опытных линуксоидов. LXF



Hymera Linux

Разработчик: Hymera Engineering
Сайт: http://www.hymera.it/en/
Цена: Бесплатная загрузка или €40—90 в зависимости от варианта

Функциональность	9/10
Производительность	8/10
Простота использования	8/10
Внешний вид	10/10
Оправданность цены	0/10

» Дистрибутив из разряда «все включено» с выдающимся визуальным оформлением.

Рейтинг 8/10

Muse 1.0.1



Известный музыкальный пакет миновал очередную веху, но конкуренция накаляется. **Дэниел Джеймс** думает: удержит ли Muse свой темп?

Вкратце

» **Аудио- и MIDI-секвенсор с программными синтезаторами, модулями DSSI и поддержкой VSTi. См. также:** *Qtractor*, *Rosegarden*.

Протокол MIDI остается универсальным стандартом дистанционной синхронизации и управления электронными музыкальными инструментами. Нынешние процессоры во много раз мощнее зачинателей эры цифровых синтезаторов, и теперь на обычной домашней машине можно эмулировать несколько виртуальных инструментов сразу.

В сфере свободного ПО существует два «долгоиграющих» MIDI-проекта: *Rosegarden* и *Muse*, плюс недавно к ним присоединился *Qtractor*. Все три приложения базируются на *Qt* и поддерживают аудиосервер Jack. Все умеют направлять MIDI-данные на виртуальные

> «В Muse имеются настройки для популярных инструментов.»

инструменты без специальной аппаратуры и поддерживают подключаемые модули, рассчитанные на работу в составе приложений-хостов.

Двоичные пакеты *Muse 1.0.1* имеются не в каждом дистрибутиве, но их сравнительно несложно скомпилировать из исходных текстов. Пользователям Debian и Ubuntu придется преодолеть пару дополнительных препятствий – подробности см. в документации.

Качество *Muse* обусловлено длительным опытом разработки: все стандартные функции секвенсора действуют как полагается. Программа не зависает и не нарушает синхронизации, из-за чего становятся непригодными некоторые другие музыкальные приложения.

» **Поддержка встроенных MIDI-функций ALSA помогает интеграции *Muse* с известными инструментами: «драм-машиной» *Hydrogen* или синтезатором *ZynAddSubFX*.**

Экран приветствия предлагает загрузить демо-проект с сайта *Muse* – вещь полезная, и новичкам бы он очень помог. Однако попытка загрузки не удалась. Правда, на сайте *Muse* есть вводные инструкции и несколько доходчивых видеоуроков.

Пользователям, знакомым с обычными MIDI-секвенсорами, интерфейс покажется привычным, включая окно Piano Roll для каждой MIDI-дорожки: в нем размечены продолжительность и частота каждой ноты. Для управления микшером и подачей звуков предусмотрены всплывающие окна.

Посредник

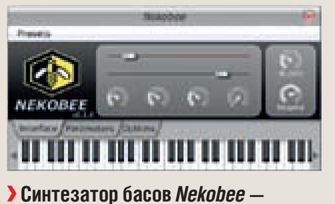
Для расстановки маркеров в главном окне секвенсора необходима средняя кнопка мыши, что весьма неудобно владельцам ноутбуков, или двухкнопочная мышь с эмуляцией средней кнопки.

Кроме пары встроенных программных синтезаторов, в *Muse* имеются готовые настройки для ряда популярных аппаратных и программных MIDI-инструментов, а это очень мило.

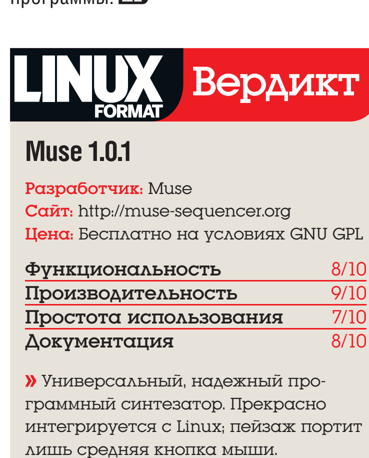
Конкуренция в данной сфере постоянно ужесточается, вот-вот появится *Ardour 3.0* со встроенным синтезатором – ПО для сочинения музыки на Linux изобильно как никогда. Так что *Muse* придется побороться за свое место под солнцем. Но, поскольку интерес к применению свободных программ в мультимедиа-проектах растет, всегда найдется достаточно пользователей и программистов для непрерывного дальнейшего совершенствования программы. LXF

Форматы синтезаторных модулей

Благодаря мосту DSSI-VST *Muse* поддерживает не только расширения программных синтезаторов DSSI, но и некоторые двоичные модули VSTi для Windows. Поскольку VSTi – проприетарный формат, и большинство модулей написаны отнюдь не для Linux, работа с ними зачастую идет по принципу проб и ошибок. Но некоторые популярные Windows-модули неплохо ведут себя в Linux при использовании *Wine*-библиотек. Существуют и двоичные модули VSTi специально для Linux, от производителей проприетарного ПО Loomer (www.loomer.co.uk) и свободного – Juced (http://code.google.com/p/juced).



» **Синтезатор басов *Nekobee* – родной Linux-модуль VSTi от Juced.**



LINUX FORMAT Вердикт

Muse 1.0.1

Разработчик: Muse
Сайт: http://muse-sequencer.org
Цена: Бесплатно на условиях GNU GPL

Функциональность	8/10
Производительность	9/10
Простота использования	7/10
Документация	8/10

» Универсальный, надежный программный синтезатор. Прекрасно интегрируется с Linux; пейзаж портит лишь средняя кнопка мыши.

Рейтинг 8/10

TonidoPlug

Полноценный Linux-сервер в корпусе от маленькой пластмассовой электровилки, говорите? **Энди Хадсон** тоже было не поверил.

Вкратце

» Крошечный Linux-сервер, предназначенный для работы с внешними USB-накопителями. См. также SheevaPlug.

Несколько месяцев назад (в LXF126/127) мы рассматривали на этих страницах миниатюрный сервер SheevaPlug. Рады сообщить вам, что это – не единственный вариант для любителей компактных компьютеров.

TonidoPlug – крошечный сервер, работающий на круто урезанной встраиваемой версии Ubuntu 9.04 (512 МБ внутренней памяти и накопитель на 512 МБ). Из портов – только Gigabit Ethernet и USB. Устройство снабжено шнуром питания со стандартной американской вилкой, которую можно снять, заменив на британскую или европейскую – если вы приобретаете TonidoPlug в официальном интернет-магазине, не забудьте выбрать правильный вариант.

> «Есть платформа для блогов, фотогалерея и сервер музыки.»



» Система общего доступа к файлам, музыкальный сервер, фотогалерея, менеджер Torrent-загрузок и прочее — все в компактном TonidoPlug на базе Ubuntu 9.04.

Подключение TonidoPlug – процесс предельно простой. Соединив его с вашей сетью через порт Ethernet, посетите сайт, который опознает новое устройство и вернет вас назад, к экрану настройки. Еще пара вопросов – и перед вами консоль администрирования: отсюда можно выполнять Torrent-загрузки и управлять мультимедиа- и рабочими приложениями. Жесткий контраст с SheevaPlug, над которым пришлось-таки потанцевать. TonidoPlug явно нацелен на пользователей, которым программирование ни к чему, а нужен просто маленький сервер для организации недорогой сети хранения данных.

Добравшись до интерфейса, вы заметите, что TonidoPlug базируется на Sheeva: уроки SheevaPlug не прошли даром.

Torrent для начинающих

Управлять загрузками BitTorrent с помощью TonidoPlug очень просто. Это один из пунктов главной страницы Tonido Admin. Откуда вы управляете загрузкой, совершенно неважно: можно указать torrent-файл напрямую или загрузить со своей машины. Можно устанавливать ограничения на загрузку и выгрузку и получать отчеты о состоянии своих торрентов почти в реальном времени. Если вы не слишком щепетильны насчет файлообмена, то можете заметить, что ваш компьютер раздает загруженное вечно. Рекомендуется ограничить подобную щедрость неким коэффициентом, после чего торрент будет остановлен.

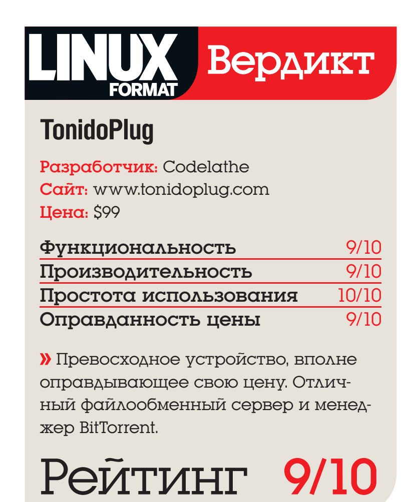


Похвальное ПО

Основа ПО – Ubuntu 9.04, но прямой доступ к консоли для серьезных доработок отсутствует. Кроме того, ряд компонентов можно добавить и удалить самим, исходя из своих потребностей. Помимо упомянутых приложений, имеется система общего доступа к файлам, платформа для блогов, фотогалерея и обязательный музыкальный сервер.

Главная же особенность устройства – простота настройки удаленного доступа. То есть администрировать TonidoPlug можно из любой точки мира: были бы Интернет да web-браузер. Например, вам позарез нужен документ, который вы составили, но забыли дома на диске, подключенном к TonidoPlug – ну так войдите в Интернет и возьмите требуемый файл. Малыш выручит вас в трудную минуту.

Это очень удобное устройство; жаль, что пришлось вернуть его в Tonido после испытаний – мы успели его полюбить. Если вам нужен компактный, но мощный сервер, то лучше TonidoPlug не найти. LXF



LINUX FORMAT Вердикт

TonidoPlug

Разработчик: Codelathe
Сайт: www.tonidoplug.com
Цена: $99

Функциональность	9/10
Производительность	9/10
Простота использования	10/10
Оправданность цены	9/10

» Превосходное устройство, вполне оправдывающее свою цену. Отличный файлообменный сервер и менеджер BitTorrent.

Рейтинг 9/10

Blender 2.5 Alpha

То, чего и ждали? Разработчики преподнесли приятный сюрприз, но постигать науку *Blender* придется с самых азов, считает **Андрей Прахов**.

Вкратце

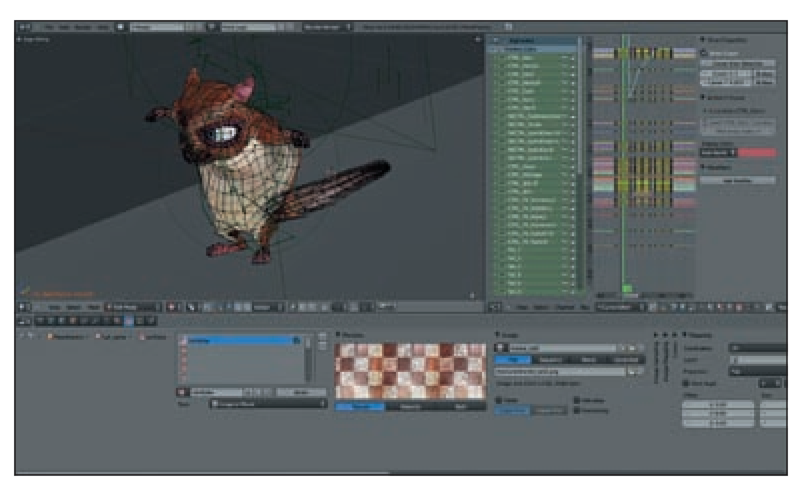
» **Популярный 3D-редактор — прямой конкурент проприетарных 3Ds Max и Maya. Среди альтернатив: Maya, Realsoft 3D, Albatross 3D.**

Многолетний опыт работы с приложением всегда накладывает свой отпечаток на восприятие новой, революционной версии. Особенно, если ее интерфейс претерпевает значительные изменения. Далеко за примером ходить не нужно: красивый, можно сказать – блестящий интерфейс KDE 4 до сих пор вызывает отторжение значительной части консервативных пользователей, предпочитающих держаться привычной третьей версии. Что же говорить, если затронута не только внешняя оболочка, но и основные принципы функционирования?! Итак, встречайте – альфа-релиз *Blender* долгожданной ветки 2.5!

И красиво, и удобно

Первый запуск программы вызывает оторопь: всё настолько непривычно... и гламурно. Да-да, вы не ошиблись, внешний вид интерфейса действительно вызывает неподдельное восхищение своей отточенностью и изяществом. Вот только опытному пользователю *Blender* все эти рюшечки навевают грустные мысли о необходимостипереучиваться.Найти,кпримеру, банальную панель Material среди имеющегося нагромождения не так-то просто. Утешает, что горячие клавиши по умолчанию и привычные названия инструментов не претерпели изменений. И тут мы с вами подошли к первому новшеству этой версии *Blender*. Теперь пользователь может поменять стандартное управление функциями программы на свое собственное. Не нравится нажимать клавишу A для выделения объектов? Без проблем – прописываем свою любимую комбинацию в настройках и радуемся жизни. К слову сказать, реализован этот механизм качественно и удобно.

Следующая возможность понравится владельцам нескольких мониторов. Не секрет, что экранного пространства при работе с 3D-графикой всегда не хватает. И хотя *Blender* позволяет иметь неограниченное количество пользовательских комбинаций активных окон, это не снимает проблему с экранной площадью. Теперь появилась возможность простым щелчком мыши отсоединить нужную область главного окна программы и перетащить на другой монитор. Важный момент – сохранение сделанных изменений в рабочем файле проекта.



› **Элегантный, блестящий интерфейс скрывает за собой недюжинные возможности программы.**

Однако следующее нововведение может поставить бывалого пользователя в тупик. Ответьте на вопрос: как проще всего добавить объект в сцену? Я уже слышу хор голосов, поющих в унисон – использовать клавишу пробела для вызова соответствующего меню. Забудьте: в новой версии программы нажатие этой клавиши вызовет непривычный для *Blender* гибрид меню с поисковой строкой. Достаточно ввести несколько букв ключевого слова, и умная программа сама предоставит подходящие варианты команд. Так, для создания примитива Cube можно ввести первые два символа. Понятно, что набор слова Add вызовет все возможные пункты с этими буквами. Особо приятно, что разработчики позаботились придать этому механизму некоторую интеллектуальность, так как подходящие значения ищутся не только благодаря совпадению букв, но и в зависимости от текущего режима редактора.

И ещё немного

Вы думаете, это всё? Ничего подобного! Напомню, что работа над концепцией *Blender 2.5* длилась много лет. Новая ветвь программы предоставляет удивительные и мощные инструменты для контроля рабочего процесса. Так, уже притчей во языцех стало повсеместное использование горячих клавиш в работе над сценой. Это ставили в укор разработчикам, так как добраться с помощью мыши до некоторых функций было весьма затруднительно. Сейчас эта проблема отпала сама собой. *Blender 2.5* обзавелся замечательной панелью Tool Shelf, которая оперативно подставляет пользователю все необходимые инструменты в зависимости от того, чем он занят в текущий момент. Редактирует модель? Пожалуйста, вот вам кнопочки от Mesh Tools. Нужно расставить объекты в сцене? Без проблем – обратите свое внимание на Tool Shelf.

А ведь пользователя ещё ждут новая анимационная система с возможностью контролировать любое свойство сцены, симулятор дыма, взаимодействующий с препятствиями, обновленный редактор частиц и многое другое. Вот только полноценно работать с этим богатством можно будет только осенью, если сроки опять не перенесут. LXF

LINUX FORMAT Вердикт

Blender 2.5 (Alpha)

Разработчик: Blender Foundation
Сайт: http://www.blender.org
Цена: Бесплатно по лицензии GPL

Функциональность	7/10
Производительность	8/10
Простота использования	5/10
Внешний вид	10/10
Оправданность цены	10/10

» Мощный кроссплатформенный 3D-редактор. Пусть и альфа, но попробуйте — он стоит того!



Рейтинг 8/10

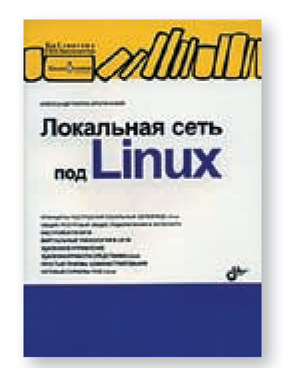
Локальная сеть под Linux

Конкретные умения и навыки или понимание принципов? **Валентин Синицын** пытается решить извечную дилемму, взяв в руки книгу.

Готов ли Linux к роли настольной ОС для массового рынка – вопрос спорный; но в том, что это оптимальный выбор для поддержания сети в малом или домашнем (а также более крупном) офисе, усомнится только завзятый скептик. Не верите – поинтересуйтесь, на какой основе собраны прошивки популярных широкополосных маршрутизаторов для сегмента SOHO. Но там-то все настраивается через удобный web-интерфейс, а что делать опытному пользователю или начинающему системному администратору, желающему сделать все самостоятельно?

Например, взять эту книгу. Она ознакомит вас с основами построения сетей и тем, что такое Linux, научит устанавливать дистрибутив и настраивать базовые системные сервисы: HTTP, FTP, DNS, NFS, *Samba*, опишет *OpenVPN* и виртуализацию (почему-то на базе *VMware Server* и *VMware Player*); а в виде бонуса приложен справочник по самым употребимым Linux-командам.

Книга, однако, не тянет на единственный или основной учебник: автор излишне склонен к графическому интерфейсу. С точки зрения пропаганды это не плохо – пользователи Windows спасены от «культурного шока»; но как освоить администрирование *Apache* без понятия о структуре файла **httpd.conf**? Вдобавок автор ориентируется на не самые новые или популярные дистрибутивы: ASPLinux, Mandriva 2008... А если читатель решит поставить Debian или Ubuntu Server? Правда, есть неплохое описание *Firestarter* – по нему уверенный пользователь Windows XP, пожелав разделить с семьей интернет-соединение со своего нового Linux-компьютера, управится за какие-нибудь полчаса. Как ни странно, довольно подробно освещено составление файлов зон *Bind*, но не упомянуто о *Dnsmasq*, который лучше вписался бы в общую схему.



› Любите книжки с картинками? Тогда эта должна вам понравиться.

Книга оставляет ощущение написанной новичком [в Linux] и для новичков: она не слишком приближает к пониманию того, как работают сетевые сервисы в Linux, но помогает быстро решить типовые задачи. LXF



Локальная сеть под Linux

Автор: Александр Поляк-Брагинский
Издательство: БХВ-Петербург
ISBN: 978-5-9775-0171-2
Цена: 187 руб.
Объем: 240 стр.

» Сгодится как введение в предмет, но не рассчитывайте, что прочитав ее, вы узнаете об администрировании Linux-сетей все.

Рейтинг 5/10

Сравнение



Каждый месяц мы сравниваем тысячи программ — а вы можете отдыхать!

Мини-дистрибутивы

По **Нику Вейчу**, меньше – значит, лучше. Идем в мир экономичных дистрибутивов.



Про наш тест...

Основной задачей теста было узнать, хорошо ли эти дистрибутивы работают со скромными ресурсами. Поэтому мы проверяли их, где это было возможно, на древнем ноутбуке Compaq с 256 МБ ОЗУ, графикой Vesa, 4-гигабайтным жестким диском и процессором Pentium на 200 МГц. Ну и, здравомыслия ради, затем они были дополнительно протестированы в виртуальной среде *Qemu* с теми же ограничениями, но на сей раз на половине 3-ГГц процессора Core 2 Duo.

Мы не делали никаких специальных тестов, кроме установки этих дистрибутивов (что уже само по себе являлось тестовой проверкой) и попыток выполнить несколько стандартных задач рабочего стола.

Наш выбор

Есть масса причин для установки на своем компьютере нетребовательного к ресурсам дистрибутива. Может, вы хотите вдохнуть новую жизнь в устаревшее оборудование. Или вам нужно нечто, способное уместиться на небольшой флэшке. Или вы затеваете запустить на рабочем столе сразу 200 виртуальных машин.

Для нас здесь важны такие вещи, как требуемый объем дискового пространства, мощность процессора, нужная для работы дистрибутива на приемлемом уровне комфорта, и усилия, необходимые, чтобы заставить его запуститься. Следует принять во внимание, что один из способов срезать с дистрибутива жирок – это отказ от скриптов и мастеров-помощников, уже принимаемых как должное. Это может неожиданно осложнить задачи, которые мы привычно считаем простыми: например, установку ПО.

> «Ваши пальцы больше oGrubeют при работе с таким дистрибутивом.»

Строгие критерии

Простой истиной является то, что ваши пальцы больше o*Grub*еют при работе с таким дистрибутивом, чем с полнофункциональным.

Формируя наш список, мы не включили некоторых претендентов либо потому, что они не поддерживают более старые процессоры, не влезают в 4 ГБ, не работают на нашем оборудовании или больше не поддерживаются (как RULE и U-Lite). Единственное исключение – Damn Small Linux: хотя и прошел уже год с момента последнего релиза, и на главной странице такая же тишина, как в офисе **LXF** в 9.30 утра в понедельник, проект все еще широко используется и столь влиятелен, что мы сочли его достойным размещения в списке.

В области дистрибутивов, нетребовательных к ресурсам, делают еще много чего, в том числе WattOS, который мы надеемся осветить в следующий раз. Мы также пробовали Zenwalk, но не справились с его запуском на системе с низкими спецификациями, которую мы себе здесь позволили. Но в остальном это легкий и умелый дистрибутив, и его стоит посмотреть, если у вас есть время.

Damn Small Linux

Оригинален и размером с кредитную карточку.

Взлет и падение Damn Small Linux – одна из историй, характерных для прекрасных концепций с хорошим воплощением. Идея была создать дистрибутив Linux, способный уместиться на мини-CD размером с кредитную карточку, то есть в 50 МБ или менее. При таком формате разработчикам, безусловно, придется поломать голову, если они хотят обеспечить работу пользователей без проблем.

DSL в основном преуспел. Основанный на Knoppix, дедушке всех Live CD, DSL слой за слоем убирает накипь, оставляя только главные функции системы. Может, он и не изобилует приложениями, но их достаточно, чтобы DSL законно носил звание настольной ОС. Отвлекитесь от неказистого интерфейса и сложного для чтения текста, и вы будете поражены богатством функциональности DSL. Текстовые редакторы, просмотрщик PDF, *Firefox* и другие полезные утилиты обеспечивают реальную и стабильную среду. Существуют дополнительные пакеты под определенные задачи, доступные для скачивания, и трудно не оценить уровень поддержки оборудования.

К сожалению, история DSL пока не имеет счастливого конца. Сообщество его разработчиков, похоже, распалось на фракции по отношению к требованиям некоторых участников, и уже прошел год с тех пор, как кто-либо из основных разработчиков хотя бы размещал новости на сайте проекта. Будущее дистрибутива представляется туманным. Мы включили DSL (в отличие от других забытых систем), потому что он еще на удивление хорошо держится на фоне конкурентов и по-прежнему широко используется. Кому требуются лишние доказательства – DSL выбран в качестве одной из немногих систем, поддерживаемых проектом boot.kernel.org (BKO).

При всем при том, с течением времени DSL, очевидно, будет становиться все более и более устаревшим, и может в конечном итоге превратиться в нечто вроде дани времени.

› На вид DSL не слишком элегантен, зато в него втиснута впечатляющая подборка приложений.

> «Вы будете поражены богатством функциональности DSL.»



LINUX FORMAT Вердикт

Damn Small Linux

Версия: 4.4.10
Сайт: www.damnsmalllinux.org
Цена: Бесплатно

» Хороший выбор для тех, у кого есть лишнее место на жестком диске и оперативная память.

Рейтинг 7/10

Crunchbang

Неофициально – Ubuntu Lite.

Задолго до появления официального проекта Ubuntu-lite, почву пробовали его предшественники, типа Xubuntu и U-list. CrunchBang ('#!', дошло?), или HashPling, как кое-кто мог бы его назвать, появился чуть позже, но до официальной поддержки проекта Lubuntu. Эта «фора», насколько мы можем судить, принесла свои результаты, и CrunchBang сегодня – «узнаваемый бренд».

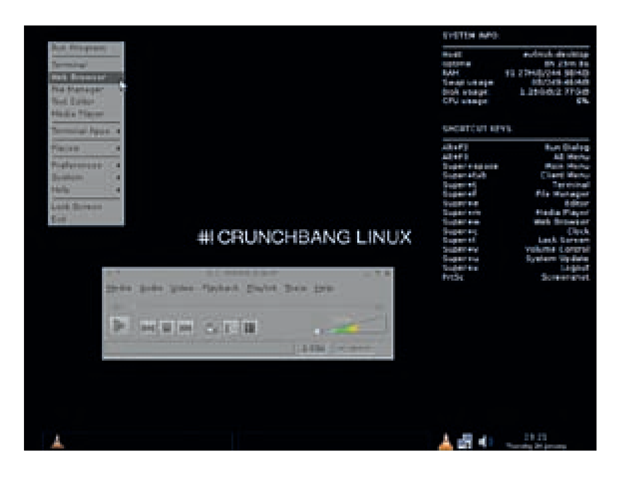
Он поставляется в нескольких вариантах, и мы решили протестировать версию Lite: она лучше вписывается в тему нашего Сравнения.

Установщик был одним из самых простых в использовании, но не сработал на нашем почтенном оборудовании – тестировали только на виртуальной машине. Видеодрайвер, как выяснилось, вызывает некоторые затруднения, поэтому ваш опыт может быть другим.

При всей облегченности, эта версия все же содержит полезные приложения, в том числе редактор *Leafpad*, *VLC* и *Firefox 3.0.11*. Одно из основных ее преимуществ – то, что дистрибутив построен на Ubuntu, причем включенный *Synaptic Package Manager* будет счастлив скачать все что угодно с репозиториев на ваш компьютер. Но установка крупных приложений не пройдет: зависимостям нужны гигабайты дискового пространства.

CrunchBang также принимает необычные, но радующие шаги в наполнении рабочего стола горячими клавишами – совершенно буквально: их список отображается на экране с помощью системного монитора *Conky*. Большей частью они используют «особую клавишу с пингвином», и не будут мешать нормальной работе.

CrunchBang невелик, элегантен и работает хорошо. Будет интересно посмотреть, что произойдет с ним, когда Lubuntu выпустят официально, но кажется, что CrunchBang имеет достаточно твердое идущее начинание для продолжения выбранного пути развития.



› Хотя он и выглядит минималистичным, ChrunchBang занимает кучу места.

> «При всей облегченности, версия содержит полезные приложения.»



LINUX FORMAT Вердикт

ChrunchBang

Версия: 9.0.4 Lite
Сайт: www.crunchbanglinux.org
Цена: Бесплатно

» Стильный, компактный и с большим количеством доступного ПО от Ubuntu.

Рейтинг 8/10

Lubuntu

Официальный мини-дистрибутив от Ubuntu.

В начале 2009 года Марк «Космонавт» Шаттлворт дал добро на проект Ubuntu, планирующий создать легковесный вариант любимого дистрибутива всего мира. Основанный на *LXDE*, Lubuntu встал на путь своего развития. И он все еще в пути. Ну, создание нового дистрибутива требует не один месяц, поэтому не будем слишком суровы. Стоит также отметить, что на момент написания текущей версией была еще альфа, так что мы не будем к ней особо придираться.

Как и большинство других представленных здесь дистрибутивов, установка сначала запускается как Live CD; это хороший способ убедиться, что система сработается с вашим оборудованием, прежде чем связываться с ее инсталляцией.

Предположение, что Lubuntu придерживается облички Ubuntu, рухнет минуту спустя после загрузки рабочего стола. Lubuntu имеет больше общего с другими дистрибутивами *LXDE*: в нижней части экрана расположена *LXPanel*, и вид более подобен KDE 3.x, а не Gnome. Состав приложений не совсем обычен: среди прочего, включены *Firefox*, *AbiWord* и *Gnumeric* – видимо, с намеком, что не все в этом дистрибутиве обглодано до костей.

Конечно, главное преимущество – доступ к репозиториям Ubuntu: это означает легкость обновления и, при желании, великое множество дополнительных пакетов для установки.

У нас-таки была пара проблем с установкой его на диск, так что данные сравнительной таблицы на стр. 21, насчет использования памяти и дискового пространства, не так уж надежны. Однако, поскольку это все еще альфа-релиз, на них так и так полагаться не приходится

За Lubuntu, безусловно, есть смысл понаблюдать. При поддержке Canonical он получит отличные кадры разработчиков, на зависть другим легким дистрибутивам.

> «Lubuntu имеет больше общего с другими дистрибутивами LXDE.»

› Что видите, то и получаете. Не больше. Но идеально для мобильных устройств.

LINUX FORMAT Вердикт

Lubuntu

Версия: Lucid Alpha 2
Сайт: http://lubuntu.net
Цена: Бесплатно

» Убунтовцы, вы удивитесь: рабочий стол быстрый и функциональный.

Рейтинг 6/10

Puppy Linux

Что это за карманный щеночек?

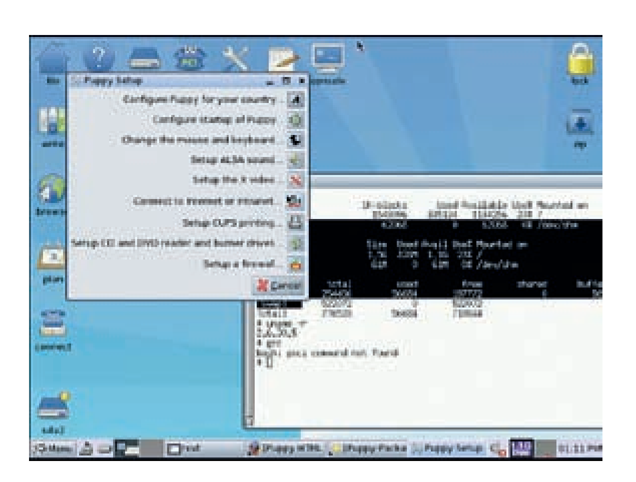
Puppy [щенок] должен, по идее, быть основан на Yellow Dog [Желтая собака], но на самом деле это абсолютно независимый дистрибутив, с начала и до конца. Его категория – средний вес: он не так обкорнан, как некоторые другие, но и не раздут до полной CD-версии. Потребление памяти – от низкого до среднего, и новейшее ядро дает хорошие шансы на поддержку оборудования, хотя поладит и с i386.

При первоначальной загрузке он работает прямо из оперативной памяти и выкатывает рабочий стол, усеянный вдумчиво выбранными приложениями. Тут же вы найдете множество полезных скриптов, например, для настройки дисплея и установки на диск, но несколько этапов все равно придется выполнять вручную. Как это часто бывает, чем меньше наворотов, тем меньше и приложений, делающих все за вас, и придется малость потрудиться.

Puppy сумел упаковать в свое малое пространство кучу программ. Графикой занимается облегченная версия *Inkscape*; есть несколько инструментов для фотокамер, *MTPaint* и *Gxine*. О просмотре web и почты заботится полная версия *SeaMonkey*, а не отдельные приложения; ну, а *Gnumeric* и *AbiWord* должны покрыть большинство офисных нужд.

Дополнительно доступны *IceWM* и *Openbox*, если вам не нравится оконный менеджер по умолчанию, а также ряд других инструментов. Конечно, имеется также и *GCC*, и вы сможете собрать собственное ПО – которое явно может понадобиться, поскольку в репозиториях хранится лишь несколько десятков приложений.

Хотя Puppy ограничен с точки зрения состава доступных программ, его можно рекомендовать еще много из-за чего: он работает как стабильный, современный дистрибутив, а места занимает всего ничего. Однако если вам необходимы конкретные приложения, пожалуй, будет лучше поискать в другом месте.



› У Puppy Linux быстрый и легко настраиваемый интерфейс.

> «Стабильный дистрибутив, а места занимает всего ничего.»



LINUX FORMAT Вердикт

Puppy Linux

Версия: 4.3.1
Сайт: www.puppylinux.org
Цена: Бесплатно

» Стабильный и надежный вариант, но доступно маловато приложений.

Рейтинг 6/10

Slitaz

Самопалит с 2007 года.

В основе легких дистрибутивов Linux нередко лежат более популярные настольные варианты, например, Debian; а Slitaz растет с нуля, начиная с 2007 года. Он один из немногих, предусматривающих не только английский язык (есть еще испанский, французский, немецкий и португальский).

Базовая установка в большинстве случаев достаточна. Браузер *Firefox 3.5*, может, и не самый легковесный, зато он дает Slitaz возможность выполнять почти любое web-приложение, что будет для многих хорошим подспорьем, поскольку самостоятельных программ не в избытке. Тем не менее, в минимальной установке есть набор полезных инструментов, в том числе *MTPaint*, читалка PDF, музыкальный проигрыватель и пара редакторов (*Leafpad* и *Nano*).

Невзирая на свою принадлежность к проектам легким и встраиваемым, Slitaz невероятным образом уместил функциональный web-сервер (*Lighttpd*) с поддержкой PHP/CGI, а также ряд других стандартных инструментов для работы с сетью (типа SSH и FTP).

При желании обогатить систему вы найдете более тысячи пакетов, доступных в онлайн-репозитории. Управляются они инструментом под названием *Tazpkg* – крошечным, но простым в использовании. Сами пакеты являются спецархивами, содержащими информацию и зависимости, так что особых проблем не будет (хотя придется ограничиться пакетами, предлагаемыми репозиторием Slitaz, или уж создавать свой собственный репозиторий самим).

Рабочий стол использует проворный и «бюджетный» оконный менеджер *Openbox*, в сочетании с рабочим столом *LXDE*, достаточно интуитивным для большинства пользователей (он родня рабочего стола KDE 3.x).

Slitaz пытается достичь многого в небольшом пространстве. Здесь нет особого выбора пакетов по умолчанию, но те, что есть, делают свое дело, причем очень быстро.

› Скрипты настройки и установщики довольно легко найти в меню.

> «Уместил полнофункциональный web-сервер с поддержкой PHP/CGI.»



Slitaz

Версия: 2.0 Cooking
Сайт: www.slitaz.org
Цена: Бесплатно

» Невероятно быстрый, обманчиво мощный и имеет свой встроенный web-сервер.

Рейтинг 9/10

Tiny Core Linux

Мельче мелкого.

Проект Tiny Core был начат в 2008 году одним из беглецов из DSL, и неудивительно, что ему присущ тот же дух – получить максимум при минимуме объема. Если чем-нибудь можно поступиться, Tiny Core сделает это самым свирепым образом, урезав базовый пакет до объема, балансирующего на грани сдачи экзамена на звание ОС Linux.

Хотя это и неплохо для желающих установить ОС на древнем оборудовании или встроить в устройство, но неизбежно означает больше работы, если вы хотите не просто загрузить ее и посмотреть на дисплей X.

К счастью, есть установщик приложений, обеспечивающий доступ к большим репозиториям пакетов TCZ, и новые программы добавить легко. Зависимости обрабатываются, но, очевидно, если вы решите установить нечто вроде *Firefox*, то дистрибутив захватит побольше места на диске. Однако в процессе работы все равно придется что-то да установить, иначе несколько системных скриптов и терминал будут вашими единственными союзниками.

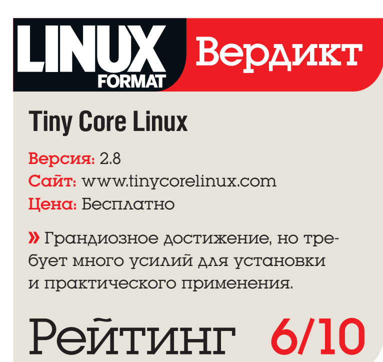
Не всегда так уж удобно иметь столь маленький дистрибутив. В частных случаях – возможно, но в общем большинство людей легко изыщут какие-нибудь дополнительные 100 МБ. Конечно, можно расширить Tiny Core, добавив приложений, но проще с самого начала задать планку повыше.

Мы отнюдь не пытаемся опорочить замечательное достижение – установку Linux всего на 10 МБ. Tiny Core способен стать отличной основой специализированного дистрибутива – снизив размер базовой системы, вы отыграете намного больше места для пользовательских программ.



› Ура! Какой быстрый запуск! А теперь что сделаем? Ой…

> «Не всегда так уж удобно иметь маленький дистрибутив.»



Tiny Core Linux

Версия: 2.8
Сайт: www.tinycorelinux.com
Цена: Бесплатно

» Грандиозное достижение, но требует много усилий для установки и практического применения.

Рейтинг 6/10

Unity Linux

Смачный потомок Mandriva.

Этот дистрибутив на базе Mandriva стремится удовольствоваться скромными ресурсами, но не хочет ходить в золушках. Пожалуй, он красивее всех своих коллег по данному Сравнению, хотя платит за это медленностью загрузки. При запуске Unity тормозит почти как полномерный дистрибутив – ему далеко до шустрых Slitaz и Tiny Core. После запуска основанного на *Openbox* рабочего стола он вполне быстр и отзывчив, как полагается нормальной системе.

Процесс установки – проще некуда: запустите графический инсталлятор, сообщите ему, где вы живете, разрешите разбить диск, как он хочет – и через пару щелчков мыши все готово. Честно говоря, это уж слишком легко: возможно, лучше бы он чуть побольше расспросил вас, куда вы хотите его установить; но на большинстве этапов есть также и возможность ручного вмешательства.

Дожидаясь, пока пройдет установка, вы всегда можете насладиться «живым» Unity, а затем перезагрузиться обратно в этот чудесный рабочий стол.

Тут вас и настигнет шок – Unity сожрал почти 1 ГБ, еще до того, как вы что-то поставили! Минимальная установка включает массу инструментов настройки, но если вам надо что-нибудь делать – просмотреть web-страницу или проиграть музыку – не обойтись без загрузки.

Удобный менеджер пакетов настроен на получение обновления и пакеты с обширных зеркал Unity, хотя и Mandriva или обычные пакеты RPM, пожалуй, не создали бы вам особых затруднений. Настройка сети прошла без проблем, и мы легкомысленно надергали всяких беспутных приложений, типа просмотрщика изображений и проигрывателя звука, за сущие секунды.

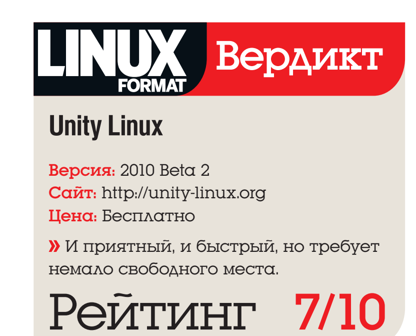
Как ни странно, по завершении установки Unity потребляет не так уж много памяти, и нам он показался бойким и легким в использовании.

> «Тут вас и настигнет шок — Unity сожрал почти 1 ГБ!»



› 5+ за внешний вид, но уж больно велик аппетит Unity на дисковое пространство.

Как и у некоторых других рассмотренных нами дистрибутивов, это бета-версия, но, по нашему впечатлению, она уже готова к полному релизу.



LINUX FORMAT Вердикт

Unity Linux

Версия: 2010 Beta 2
Сайт: http://unity-linux.org
Цена: Бесплатно

» И приятный, и быстрый, но требует немало свободного места.

Рейтинг 7/10

VectorLinux

Силен мощью Slackware.

Основанный на Slackware, VectorLinux изначально был нацелен на небольшой размер, автономность и простоту в установке и использовании. Он стартовал в 2000 году и прошел много ступеней развития, породив несколько ответвлений (SOHO, Deluxe, Standard, Light), нацеленных на конкретное применение. Мы тестировали версию Light, хотя и она тянет на полный CD.

Да, 617 МБ – явно тяжелее коллег по Сравнению. Даже если отказаться от дополнительных пакетов, установка Light требует 1 ГБ дискового пространства, поэтому не удивительно, что выбор приложений здесь широк. Инструменты разработки и исходные тексты ядра в принципе необязательны, и устанавливать их на менее чем 4-гигабайтный диск мы бы не рекомендовали, если вы хотите иметь некоторое пространство для подкачки (существенное для машин с малым ОЗУ) и место для хранения файлов.

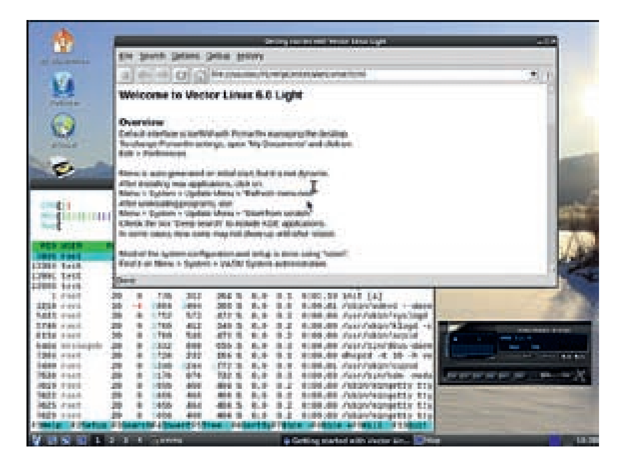
С точки зрения выбора приложений, упор сделан на Интернет и мультимедиа. Web-браузеров четыре штуки, а для офиса – только *Leafpad*, *Pathetic Writer* и *Siag Office*, плюс *MTPaint* для графики.

Ветераны инсталляции, поднаторевшие в этом за до-Ubuntu'ский период, установят VectorLinux легко. При этом путешествии в историю, на основе *Curses*, вам осмелятся задавать всякие вопросы, да еще хотят, чтобы вы сами разбили и отформатировали диск!

VectorLinux вовсе не так уж плох: просто он не особо вдохновляет. У него самый большой из всех загрузочный образ, он съедает больше всех места на диске, а работа в нем не являет никаких прорывов. В некотором смысле, вы могли также взять любой нормальный дистрибутив.

Интерфейс может показаться странным, и настроек в нем немного, но через некоторое время он становится обманчиво легким в использовании.

> «VectorLinux не плох: просто он не особо вдохновляет.»



› На фоне запросов VectorLinux даже Unity выглядит умеренным.



LINUX FORMAT Вердикт

VectorLinux

Версия: 6.0 Light
Сайт: www.vectorlinux.com
Цена: Бесплатно

» В общем, неплохой, если у вас есть лишнее место на жестком диске и достаточно ОЗУ.

Рейтинг 5/10

Мини-дистрибутивы

Вердикт

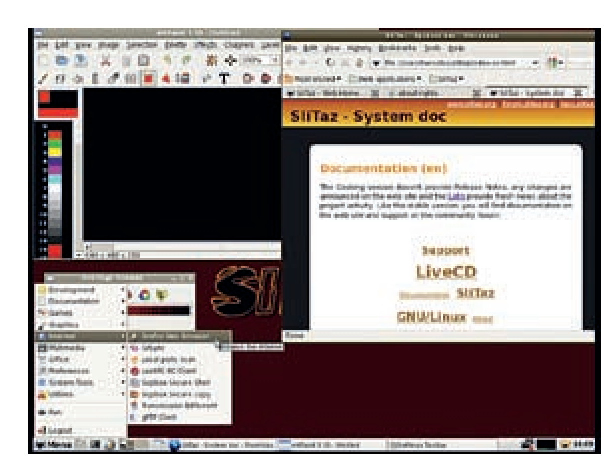
Slitaz 9/10

Мы надеемся, что показали вам: мир легковесных дистрибутивов интереснее, чем вы представляли. Ваш личный выбор зависит от оборудования, на котором вы хотите запустить «малютку», и от способа его применения.

Дистрибутивы на основе Ubuntu интересны, особенно зарождающийся Lubuntu, главным образом благодаря малым размерам и перспективе установки приложений из огромного мира Ubuntu. Однако мы-то искали дистрибутив для работы без проблем на скромном оборудовании. Особого упоминания заслуживают DSL и Tiny Core, завоевавшие сегмент малюток. Удивительно, насколько удобной может быть система, занимающая так мало места на диске – сравнимо с объемом ваших отпускных фоток. Puppy Linux и Unity отличаются простой использования, хотя последний был немного более отлаженным (и большим).

Но в контексте нашего Сравнения может быть только один победитель, и это Slitaz. Он быстр, нетребователен с точки зрения памяти и поставляется с разумным выбором приложений. Отсутствие возможности легко установить другое программное обеспечение помимо пакетов в собственном формате Slitaz – один из его немногих недостатков, но благодаря быстрому, легкому рабочему столу Slitaz трудно превзойти.

Все протестированные дистрибутивы либо инсталлировались с Live-версии, либо предусматривали таковую; перед установкой проверьте совместимость вашего оборудования. Не всегда можно уверенно заявить, что крупнейшие дистрибутивы заодно и наиболее совместимы – всяко бывает; но наши конкурсанты должны обеспечить базовую функциональность (графика, клавиатура, мышь и проводная сеть). Если ваша машина – ноутбук, ждите разного рода трудностей. Многие части ноутбука не то, чем кажутся, по крайней мере, с точки зрения драйверов ядра. LXF



› Slitaz коронован за простоту использования и скорость при малых ресурсах.

Обратная связь

Правы ли мы были, отдав предпочтение Slitaz? Нет ли какого-то другого компактного дистрибутива, который мы упустили из виду? Присылайте нам ваше мнение по электронной почте на адрес letters@linuxformat.ru.

Сравнительные характеристики

Название	CrunchBang	Damn Small Linux	Lubuntu	Puppy Linux	Slitaz	Tiny Core Linux	Unity Linux	VectorLinux
Версия	9.0.4 Lite	4.4.10	Lucid Alpha 2	4.3.1	2.0 Cooking	2.8	2010 beta 2	6.0 Light
Сайт	http://crunchbanglinux.org	www.damnsmalllinux.org	http://lubuntu.net	www.puppylinux.org	www.slitaz.org	www.tinycorelinux.com	http://unity-linux.org	www.vectorlinux.com
Основан на	Ubuntu	Knoppix	Ubuntu	—	—	—	Mandriva	Slackware
Размер образа, МБ	427	50	362	105	31	10	232	617
Размер после установки, МБ	1,206	82	1,736	328	149	16	997	2,365
Использование ОЗУ Live (1)	75	21	88	64	136	22	67	n/a
Использование ОЗУ при загрузке с диска (1)	62	21	66	56	53	22	39	56
Менеджер окон	Openbox/LXDE	JWM	Openbox/LXDE	JWM	Openbox/LXDE	JWM/FLTK	Openbox/LXDE	IceWM/JWM
Ядро	2.6.28	2.4.31	2.6.32.11	2.6.30.5	2.6.30.6	2.6.29.1	2.6.31.9	2.6.27.12
USB	✘	✔	✔	✔	✔	✔	✘	✘
CD	✔	✔	✔	✔	✔	✔	✔	(2)
Жесткий диск	✔	✔	✔	✔	✔	(3)	✔	✔
Мин. процессор	i386	i386	i486	i386	i386	i386	i586	i486

(1) Хотя в общем можно считать, что чем меньше используется памяти, тем лучше, некоторые дистрибутивы работают более эффективно при максимуме допустимых ресурсов.
(2) VectorLinux Standard и другие версии доступны как Live CD.
(3) Нет скрипта установки, но есть инструкция, как сделать это вручную.

Linux годен!

ИТ-менеджеры! Хватит тратиться на лицензии и поддержку: переводите офис на свободное ПО...



Многие покупают LXF, потому что им еще предстоит погрузиться в чудесный, волшебный мир открытых возможностей свободного ПО. Они хотят, чтобы их убедили. А так как Linux и свободное ПО уже сделали себе имя в мире бизнеса, все больше людей проверяют жизнеспособность перехода на открытые аналоги программ для дома или небольшого офиса.

Независимо от вашего отношения к рабочему столу Linux, именно в этой области он может оказать реальное влияние – как по деньгам, так и по производительности, и любой малый или домашний офис имеет шанс обрести свободу, просто заместив одно приложение одним или двумя открытыми аналогами.

Традиционно это вотчина Microsoft, дорогостоящего и подчас раздражающего программного окружения, где, чтобы удержаться на плаву, нужно вновь и вновь закупать очередную версию. Свободное ПО предлагает выход из этого порочного круга, и, что более важно, замену ему. Linux и свободное ПО дадут вам глоток свежего воздуха, причем даже не потребуется очертя голову нырять в другую операционную систему: открытые кросс-платформенные программы позволят попробовать альтернативы еще до глобального перехода.

На распутье

Многие пользователи обнаружат, что свободные аналоги программ выглядят знакомыми – и дизайном, и функциями, и на следующих страницах мы перечислим наиболее важные из них и попытаемся по дороге учесть все соображения и подводные камни.

То, что мы смогли заполнить эти страницы самыми разными альтернативами для многих задач и приложений – верный признак успешности свободного ПО, но мы хотели копнуть глубже, чтобы возможные пользователи чувствовали себя уверенно при переходе или обсуждении потенциала с теми, кто принимает решения. Когда вы дочитаете статью, не подбросить ли вам экземпляр журнала боссу на стол?

Свободный офис

О том, как мы перестали тратить деньги и полюбили свободные программы.

Важная особенность свободного ПО – независимость от используемой платформы. Если программа достаточно популярна, она наверняка портирована на вашу систему. Такие оплоты свободного ПО, как *OpenOffice.org*, *Firefox* и *GIMP* работают как под Windows, так и в Linux.

Это значит, что даже если вас пугает переход на Linux, для начала вы можете мирно дублировать программу-другую из вашего обычного арсенала и пару недель приглядываться, как идут дела. Обретя уверенность, рассмотрите замену другой программы, а может, и третьей, пока не осознаете, что Linux – не обязательно прыжок в неизведанное, каким он казался ранее.

В мире малых офисов, видимо, преобладают Microsoft и Adobe, и ни один из них не предпринял серьезных попыток портировать свои тяжеловесные продукты на рабочий стол Linux. Здесь и пришлось больше всего потрудиться разработчикам свободного ПО: они создали открытые альтернативы для основных коммерческих приложений от обеих компаний.

Мир свободного ПО полон альтернатив, так как разработчикам нравится свобода выбора. А поскольку код этих программ открыт, то после изобретения одной программой новой разновидности велосипеда вскоре можно увидеть, что ее конкуренты в мире открытого ПО ухватили суть и обеспечивают те же возможности. Многие разработчики отзывчивы к запросам на функции и личной переписке, чего уж точно не бывает ни в Adobe, ни в Microsoft. И, конечно же, если у вас или ваших коллег достаточный навык программирования, вы можете сами что-то поменять и тем самым внести вклад в сообщество.

На следующих страницах мы постараемся несколько раз подчеркнуть одну мысль. Быть может, вы и не найдете полного соответствия с приложениями, привычными для вас в проприетарном мире, но в широком диапазоне случаев этот дефицит не сделает большой разницы. Очень немногие офисные работники затрагивают расширенные возможности коммерческих приложений, и даже если вы принадлежите к меньшинству, использующему функции программ типа *Microsoft Office* на полную катушку, у нас есть решение, которое позволит сохранить ваши старые приложения в Linux.

Wine

Если вы все-таки переведете свой офис на Linux, останется два способа запускать стандартные программы Windows. Первый – *Wine,* среда для исполняемых файлов Windows на рабочем столе Linux. Приложения типа *Quicken* и старых версий *Office* запустятся без труда, а более новые версии можно заставить работать, немного «поработав напильником». Также есть коммерческое решение на базе *Wine*, под названием *CrossOver Office*. Оно запускает самую свежую версию *Office* от Microsoft, а деньги, которые вы заплатите за него, так или иначе вернутся разработчикам свободного ПО. За поддержкой в Linux популярных российских бизнес-приложений имеет смысл обратиться к Wine@Etersoft. Главное, что не будет нужна лицензия на операционную систему, а только на используемые вами программы. Здесь мы немного забегаем вперед: хотя приятно знать, что все работает, вы увидите, что тема поддержки совместимости с Windows утратит остроту по мере привыкания к инструментам, предлагаемым Linux.

> «Тема поддержки совместимости с Windows утратит остроту по мере привыкания к Linux.»

»

Открытые альтернативы

Тип приложения	Windows	Кросс-платформенный	Сайт
Текстовый процессор	Microsoft Word	Writer	www.openoffice.org
Электронные таблицы	Microsoft Excel	Calc, Gnumeric	www.openoffice.org, http://projects.gnome.org/gnumeric
Программы для презентаций	Microsoft PowerPoint	Impress	www.openoffice.org
Настольные базы данных	Microsoft Access	Base	www.openoffice.org
Редактор изображений	Adobe Photoshop	GIMP	www.gimp.org
Векторный редактор	Adobe Illustrator	Inkscape, sK1	www.inkscape.org, www.sk1project.org
Издательские системы	QuarkXPress	Scribus	www.scribus.net
Web-дизайн	Adobe Dreamweaver	Kompozer	http://kompozer.net
Создание музыки	Digidesign Pro Tools	Ardour	http://ardour.org
Редактор звука	Steinberg Wavelab	Audacity	http://audacity.sourceforge.net
Web-браузер	Internet Explorer	Mozilla Firefox	www.mozilla.org
Электронная почта	Outlook Express	Mozilla Thunderbird, Claws Mail	www.mozilla.org, www.claws-mail.org
Менеджер проектов	Microsoft Project	KPlato, OpenProj	www.koffice.org, www.openproj.org
Просмотрщик PDF	Adobe Acrobat	Evince, Okular	http://projects.gnome.org/evince, http://okular.kde.org
Финансы	Intuit Quicken	GnuCash, KMyMoney	www.gnucash.org, http://kmymoney2.sf.net

Текстовый процессор

Программа, которой нужна совместимость как никакой другой.

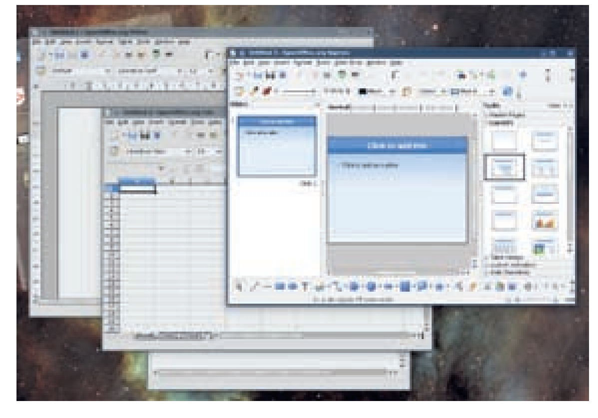
Существует один вопрос, который выходит во главу угла при определении, сможет ли бизнес мигрировать на открытое ПО и какие программы следует использовать. Это проблема совместимости с почтенным офисным пакетом от Microsoft.

Форматы файлов от Microsoft стали стандартом: они применяются везде, от заметок наскоро и внутренних почтовых сообщений до официальных документов, книг и публикаций. Расширения файлов DOC, DOCX и XLS стали синонимами обработки текста и электронных таблиц, как PPT – для презентаций, а PST – для почтовых архивов. Каждый из этих форматов принадлежит Microsoft и управляется им, и самый важный аспект в поиске альтернативы – чтобы формат замены работал наиболее прозрачно и пользователь не опасался за совместимость.

К счастью, для широкого круга пользователей и задач имеются открытые альтернативы, совместимые почти на 100 %, и на них можно перейти без малейших проблем. Существуют свободные офисные программы, способные считывать и записывать файлы в форматах, совместимых с Microsoft – то есть те, кому вы посылаете документы, даже ничего не заподозрят, если вы сами не признаетесь, что вы не платили за используемую программу ни гроша и что она создана сообществом хакеров. И это приложения столь же мощные, как их проприетарные собратья, и способны удовлетворить даже самым загадочным и экзотическим потребностям форматирования.

> «Лучший выбор для совместимости с MS Office – OpenOffice.org.»

Но будьте осторожны. Мы сказали «*почти* на 100 %» не без причин, и опытные пользователи *Microsoft Office*, особенно любители применять макросы Visual Basic в электронных таблицах, заметят разницу. Для большинства из нас эти мелкие отличия не повлияют на работу или на способ делиться файлами с другими, однако лучше иметь реальные представления о том, чего ожидать от альтернатив, и нет ничего, что было бы 100 % совместимо с *Microsoft Office*. Лучший выбор для совместимости с *MS Office* – пакет приложений по имени *OpenOffice.org* (*OOo*).



› **Свободный *OpenOffice.org* – полноценный офисный пакет с текстовым процессором, электронными таблицами, программами для презентаций, базой данных и редактором формул.**

Для офисных программ это сходное с интернет-адресом имя звучит довольно странно, но название *OpenOffice* было уже занято, так что к нему ради отличия добавили .org. Зато вы уж точно не забудете адрес сайта проекта.

OpenOffice.org

Первоначально проект не планировался как открытый. В начале 1999 это был проприетарный офисный пакет наподобие Microsoft'овского. Позже в том же году один из конкурентов Microsoft, Sun Microsystems, купил проект и раздавал его бесплатно. Через год код программы был также сделан свободным, что открыло проект для изменения всеми желающими. С тех пор *OpenOffice.org* находится в состоянии активной разработки, и стал главным оружием в битве за свободную альтернативу *Office* для Windows, OS X и рабочих столов Linux. Текущий релиз версии 3.2 более чем пригоден для выполнения повседневных офисных задач и служит идеальной заменой коммерческих конкурентов.

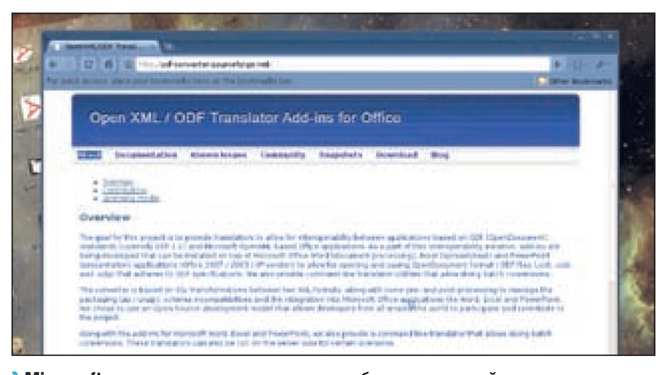
Для большинства из нас, наиболее часто (а то и единственно) используемый компонент офисных пакетов – это текстовый процессор. В *OpenOffice.org* он прозаически назван *Writer*, и если вы пользовались такими программами с тех пор, как они выбра-

Open Document Format

Создание переносимого формата файлов для обработки текстов оставалось горячей темой несколько лет. Сейчас большая часть свободных программ предлагает по умолчанию нечто под названием ODF – Open Document Format. Это файл XML с открытой спецификацией, и ее может взять и реализовать любой разработчик. Поскольку спецификация полностью открыта, разработчикам легко воссоздать в своих приложениях любые аспекты документа, не прибегая к методу проб и ошибок.

Для сравнения, форматы Microsoft обычно закрытые, и чтобы портировать их в функционал приложения, приходится выполнять обратный инжиниринг файлов, дешифруя их структуру. Создается ситуация, когда свободные программы все время пытаются догнать форматы Microsoft, а 100 % совместимости нет, поскольку на горизонте уже маячит новая версия.

У открытых форматов такой проблемы нет, потому-то Open Document Format и принят как стандарт ISO, несмотря на конкуренцию со стороны другого открытого формата, спонсируемого Microsoft (OpenXML). Если вы захотите экспортировать ODF-документ из документов Microsoft, воспользуйтесь преобразователем OpenXML/ODF, он имеется на SourceForge: http://odf-converter.sourceforge.net.



› **Microsoft частично спонсирует проект, обеспечивающий поддержку формата Open Document в ее продуктах.**

лись из опаринского бульона приложений DOS, у вас не будет проблем с его применением.

Вы пишете слова на странице, пользуетесь табуляцией и маркерами для форматирования, затем выводите результат на принтер или сохраняете его в файл. Если вам надо привыкать к новейшему поколению продуктов Microsoft, вы испытаете облегчение, увидев, что пользовательский интерфейс *Writer* прост в понимании, а это может стать реальным преимуществом, если вы также обдумываете обновление до *Microsoft Office* 2007.

Загрузка документов *MS Word* ничуть не сложнее обычного открытия файла в соответствующем диалоге *Writer*. Он распознает и формат DOC, и более новый DOCX; и хотя вы получаете предупреждение, что часть форматирования при переходе к *OpenOffice.org* теряется, разницы многие не ощутят. Если ваш документ и вправду стал выглядеть иначе, есть много способов повысить совместимость.

» Используйте стандартные шрифты

Откройте *OpenOffice.org* в Linux и *Word* в Windows – самое большое отличие между ними будет в используемых шрифтах: наборы у обоих разные. Однако вы добьетесь одинакового вида документов, придерживаясь шрифтов Sans и Serif, выбранных по умолчанию при первом запуске редактора. В Windows это означает применение MS Sans и Serif, а пользователи Linux могут выбрать у этих гарнитур версии «Liberation» или «Free».

» Делайте документы попроще

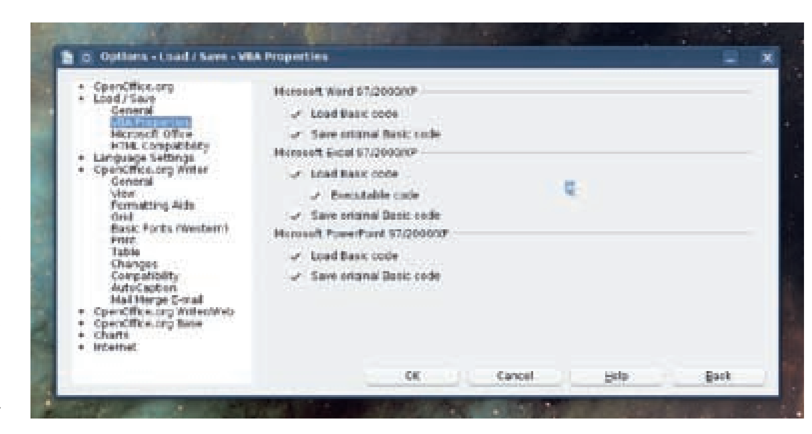
Это выглядит общим местом, но чем меньше объектов форматирования на странице, тем меньше вероятность нарваться на проблемы. Изобилие таблиц или сложные заголовки, а также определенные типы изображений и экзотические шрифты могут внести путаницу. Если что-то не является абсолютно необходимым – например, пользовательский цвет фона таблицы – то ради совместимости лучше от этого отказаться.

» Рассмотрите другие утилиты

Если вы посылаете кому-то сложный информационный бюллетень, подумайте: а является ли текстовый процессор действительно лучшей утилитой для этой работы? Вы добьетесь большей кросс-платформенной совместимости, перейдя на настольные издательские системы наподобие *Scribus* или редактор векторной графики типа *Inkscape*: оба они имеют кросс-платформенные версии, и пользоваться ими может каждый.

» Выбирайте подходящий формат

Хотя зачастую DOC и DOCX являются суровой необходимостью, все-таки, если вам нужен только голый текст, используйте нечто попроще: например, RTF или даже простой текст. С другой стороны, если не предполагается, что получатель будет править текст, наилучшим выбором может оказаться PDF, экспорт в который обеспечивается *OOo* «из коробки».



» В окне Сервис > Параметры [Tools > Options] во *Writer* скрыто множество настроек: они улучшают совместимость, и обычно включены по умолчанию.

Альтернативы для Writer

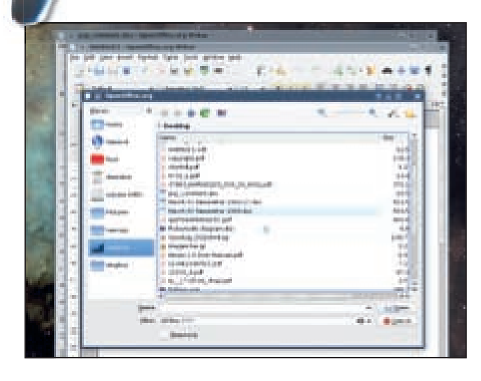
Writer – прекрасная программа, и, возможно, самая лучшая для общего использования. Однако есть и альтернативы, и если вы считаете, что интерфейс *Writer* слишком тяжелый и переполнен функциями, стоит поискать другие программы. Вы удивитесь, но существует не одно приложение, совместимое с форматами Microsoft (пусть и в меньшей степени).

Вторым по популярности текстовым процессором является *AbiWord*, и если он не установлен по умолчанию, его легко можно получить через менеджер пакетов вашего дистрибутива. *AbiWord* запускается быстрее и работает проворнее *OOo*. Ему не нужно столько зависимостей, и он нетребователен к системным ресурсам. Вы заметите это, как только откроете его из вашего меню.

В *AbiWord* есть все необходимое для работы с текстом, и освоиться в нем быстро. Имеются функции и посложнее: например, проверка грамматики, утилита для набора формул и ссылки на такие онлайн-ресурсы, как перевод на другой язык. Более новые версии включают меню Collaborate [Совместная работа], где можно создавать учетную запись и работать над документами в реальном времени совместно с другими людьми из вашего офиса или в Интернете. Это прекрасное дополнение, и в *Microsoft Word* так просто этого сделать нельзя. »

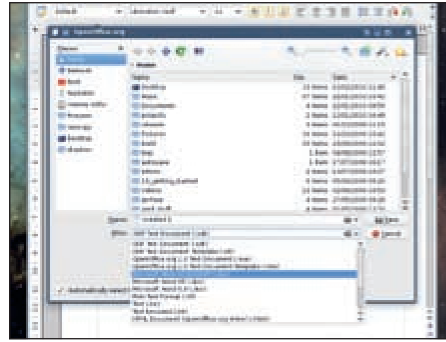


Шаг за шагом: Импорт и экспорт файлов Word в OOo



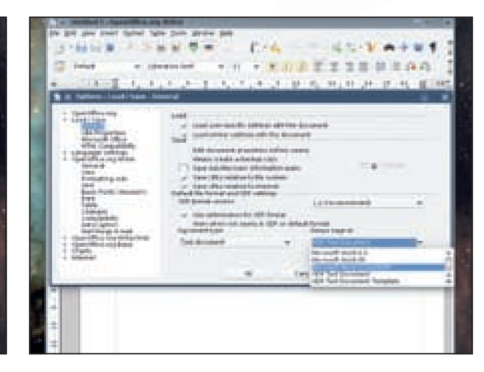
1 Загрузка файлов

Doc-файлы загружаются прямо из диалога открытия файлов *Writer* без всякой настройки. Просто игнорируйте сообщение о возможной потере информации.



2 Сохранение файлов

При сохранении файла можно выбрать формат **.doc** или более новый **.docx**. Рекомендуем использовать **.doc**, для большей совместимости.



3 Формат по умолчанию

Можно заставить *Writer* при сохранении всегда использовать формат **.doc**, из панели Параметры > Загрузка/Сохранение > Общие [Options > Load/Save > General].

Администрация офиса

После текстового процессора идет квартет программ, нужных в любой конторе.

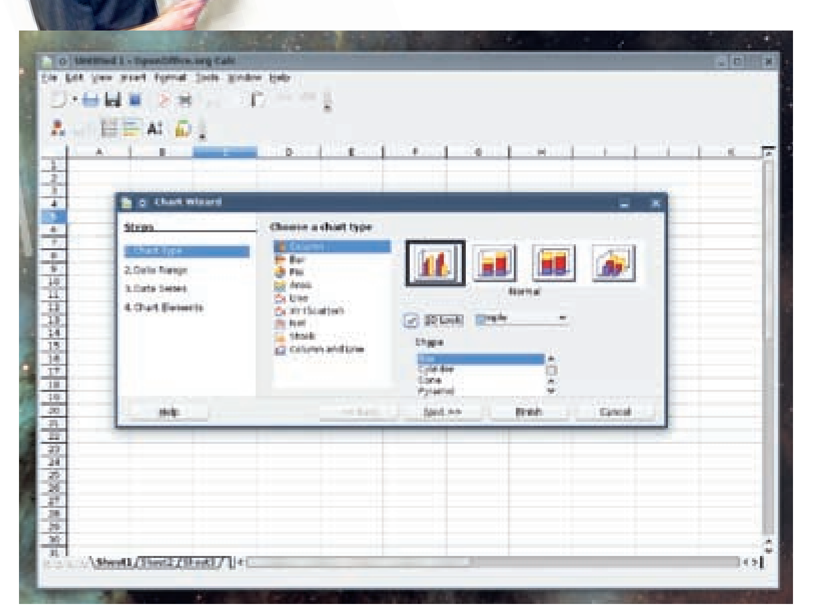
Типовая офисная работа складывается не только из написания писем, и многие офисы используют единый комплект приложений для большинства задач. Которое из них вам ближе, очевидно, зависит от рода вашей деятельности, но вряд ли вы совершенно не сталкиваетесь с электронными таблицами или почтовым клиентом, а то и с созданием презентации. Такие приложения – рабочие лошадки, без которых офис немыслим, и хотя это не непременно лучшие программы в мире, они существенны, на какой бы платформе вы ни трудились.

Всякий, кто работает с числами, когда-нибудь да применял электронные таблицы. Поскольку они вторые по важности после текстового процессора, выбор весьма широк. Самый очевидный вариант (и с наилучшей поддержкой) – программа обработки электронных таблиц из пакета *OpenOffice.org*. Имя ей *Calc*, и по понятным причинам она похожа на *Excel* от Microsoft. Если вы пользовались *Excel*, проблем с применением *Calc* у вас не будет. И, что важнее, *Calc* сумеет загрузить и сохранить файлы из ваших наработок.

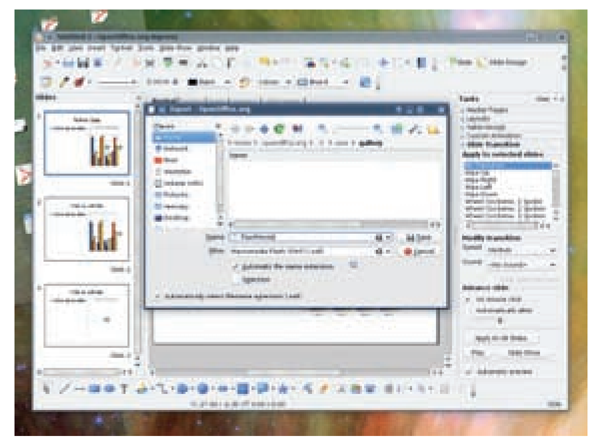
Однако электронные таблицы *Excel* бывают сложны, и, в частности, дополнены так называемыми VB-скриптами и макросами. Это быстрые и неформальные куски программистской логики, расширяющие функции электронной таблицы, когда предоставляемого по умолчанию вам недостаточно. Они обычно применяются при обработке больших наборов данных, или для ввода/вывода из специальных источников.

Язык как таковой – производное Visual Basic от Microsoft, и в программах типа *Excel* его называют Visual Basic for Applications, или VBA. Обеспечить совместимость с VBA в программах, не разработанных Microsoft – трудная задача. Она добавляет новый уровень сложности и дает новую цель, к которой нужно стремиться разработчикам.

И если ваша электронная таблица опирается в своей работе на VBA, с поиском открытой альтернативы, где она сохранит 100 % своей функциональности, возникнут проблемы. Например, в *OpenOffice.org* совместимость с VBA еще только разрабатывается. Проблема в том, что люди пишут VBA-скрипты для *Excel*, не рассчитывая использовать их в других приложениях.



› Большинство задач электронных таблиц выполнимо в *Calc* из *OpenOffice.org*.



› Да, презентации — штука нудная, но в *OOo Impress* хотя бы не надо за них платить.

Go-oo

Но не все потеряно. VBA-совместимость в основном работает хорошо, и есть специальный проект, нацеленный на повышение совместимости с *OpenOffice.org* – он называется *Go-oo*.

По сравнению с версией *OpenOffice.org*, доступной по умолчанию, *Go-oo* содержит массу улучшений. Им не нашлось места в главном проекте, и они поддерживаются под названием *Go-oo*. Сюда входят повышенная производительность, поддержка SVG, трехмерные переходы в презентациях, линейная оптимизация в электронных таблицах и лучшее взаимодействие с *Excel*, а что самое главное – поддержка макросов VBA.

Если вы установили пакет *Go-oo* в дистрибутиве Linux, то увидите, что большая часть этих улучшений уже попала в вашу версию, однако если вы пробуете свободные программы на платформах Windows или OS X, вам, скорее всего, понадобится установщик из http://go-oo.org, а не с официального сайта *OpenOffice.org*

Презентации

На программы составления презентаций в последние годы принапегли. Кто из нас не изнывал, глядя, как коллега битый час щелчками гоняет страницы с сухой статистикой и еле ползущими графиками? *PowerPoint*, программа Microsoft, ставшая синонимом данной деятельности, вошла за это в пословицу. Но это не повод выкидывать презентации и программы для презентаций на помойку. Они все еще играют важную роль.

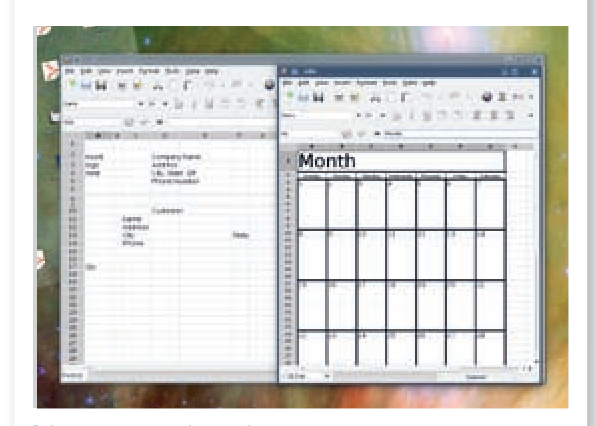
Эквивалент для *PowerPoint* в *OpenOffice.org* называется *Impress*, и в нем немало солидных преимуществ над продуктом Microsoft. Во-первых, благодаря участию сообщества существуют десятки неплохих шаблонов, которые свободно можно брать для оживления стиля вашей презентации. Зайдите на http://templates.services.openoffice.org и нажмите на Impress в облаке тэгов слева. Вы увидите, что там есть из чего выбрать, и все это загружается в приложение через стартовый мастер, появляющийся при запуске.

Открытые альтернативы

Подлинная красота открытого ПО в том, что всегда есть альтернативы, и если вам не нравится работа одного приложения, велика вероятность, что найдется другое, подходящее. Существует несколько программ для электронных таблиц, способных конкурировать с *Calc* в плане удобства, хотя до такой же совместимости с *Excel* им далеко.

Две наиболее популярных из них называются *KSpread* и *Gnumeric*. *KSpread* – часть пакета *KOffice*, группы программ, идеально вписанных в рабочий стол KDE; она включает текстовый процессор, редактор графики и утилиту для презентаций, а также электронные таблицы. В целом пакет неплохой: текстовый процессор мы бы рекомендовать не стали, а вот *KSpread* – достойная замена *Calc*, если вам нужна легковесная числодробилка.

Более зрелое приложение – *Gnumeric*; существует также версия для Windows, что повышает его шансы как альтернативы для *Calc* и *Excel*. Как и *KSpread*, оно умеет импортировать электронные таблицы многих других программ, включая *Excel*; это быстрое, эффективное приложение, удовлетворяющее множество людей. С точки зрения электронных таблиц, эта альтернатива эквивалентна *AbiWord* мира текстовых процессоров, и ее безусловно следует испытать, если *Calc* вам не по душе.
http://projects.gnome.org/gnumeric



› Электронные таблицы *Gnumeric* содержат ряд полезных шаблонов для основных числовых задач.

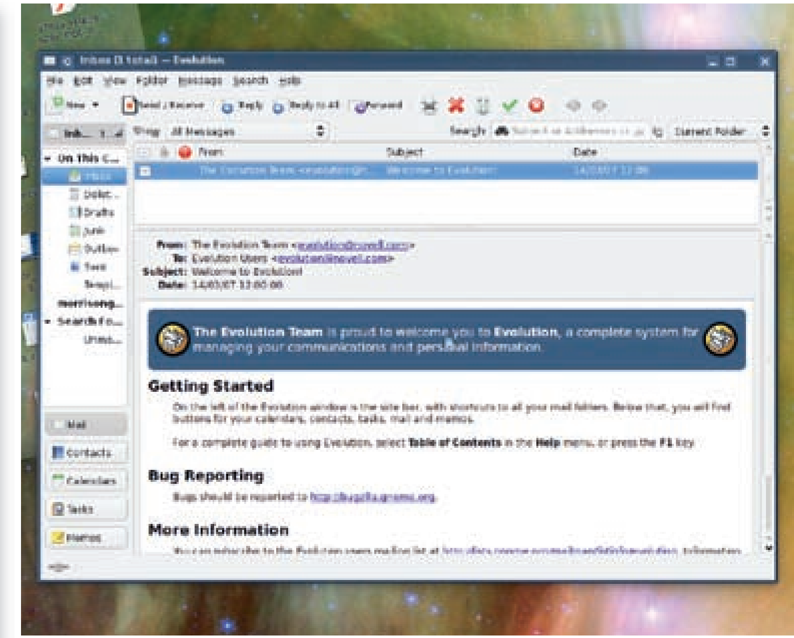
Диапазон форматов файлов, экспортируемых из *Impress*, также широк. Например, можно создавать сайты в HTML, а также PDF-документацию, если выберете пункт меню Файл > Экспорт [File > Export]. Но самый хитроумный, наверное, экспорт Flash-клипа: он сохранит все ваши тщательно отработанные переходы и графику и позволит разместить файл онлайн, чтобы другие смогли посмотреть его автоматически. Если ваши презентации удались, вы станете звездой YouTube.

Impress также хорошо совместим с *PowerPoint*, и сохраняет и загружает файлы формата PPT без осложнений. Менеджер презентаций из *KOffice*, *KPresenter*, тоже неплох, но его выбор скромнее, чем у *OpenOffice.org*, а совместимость с *PowerPoint* гораздо хуже, чем у *Impress*. Над программой для презентаций работают несколько проектов, включая *MagicPoint* и *KeyJNote*, но ни один из них не сравним с *Impress* по функциональности – их следует брать в расчет только при полном омерзении к подходу *PowerPoint* и *Impress*.

> «В Go-oo масса улучшений против стандартной версии OOo.»

Электронная почта

В любом офисе на базе Windows никуда не деться от *Outlook*. Многие люди считают его истинным лицом электронной почты, точно так же, как *Internet Explorer* – единственным вариантом выхода в Интернет. Как и в случае с *IE*, существуют серьезные открытые альтернативы, способные дать свежую замену выскальзывающим панелям и интерфейсу все-в-одном, предлагаемым *Outlook* и *Outlook Express*.



› Управляйте своей почтой, контактами, задачами и календарем из *Evolution*.

То, что вам больше подходит, сильно зависит от способа доставки вашей почты. Если вы пользуетесь *Microsoft Exchange*, ситуация усложняется. Решения, обеспечивающие адаптеры к Exchange для локальных клиентов или задействующие IMAP на сервере, существуют, но требуют определенной мороки. Будет проще попросить новообращенного линуксоида использовать web-интерфейс *Exchange*, пока не будет найдено лучшее решение.

Однако в небольшом или домашнем офисе вы, скорее всего, забираете почту из интернет-службы третьей стороны, либо как часть пакета услуг хостинга, либо от онлайн-провайдера типа Google или Yahoo. Тогда вы, скорее всего, пользуетесь для связи с этими удаленными серверами протоколами IMAP или POP3, а они широко поддерживаются большинством почтовых клиентов, невзирая на платформу.

В категории кросс-платформенных программ недавно вышел новый *Thunderbird 3*. Созданное теми же людьми, что принесли успех *Firefox*, это зрелое и стабильное приложение, расширяемое множеством дополнений, как и сам *Firefox*. В последнем релизе появились вкладки для просмотра почты и всесторонний поиск, и в качестве замены *Outlook Express* это революционный вариант.

Тем, кому обязательно нужно держаться поближе к *Outlook*, по функциональности и дизайну лучше всего подойдет почтовый клиент *Evolution* (http://projects.gnome.org/evolution). Это выбор по умолчанию в дистрибутивах на базе Gnome, и он включает менеджер контактов и календарь для полного соответствия статусу персонального органайзера. Желая добиться аналогичного эффекта на платформе Mozilla, попробуйте *Lighting* (расширение для *Thunderbird*) или *Sunbird* (самостоятельное приложение-календарь). »



Печать и дизайн

Дизайн экстра-класса отлично создается и в открытых программах.

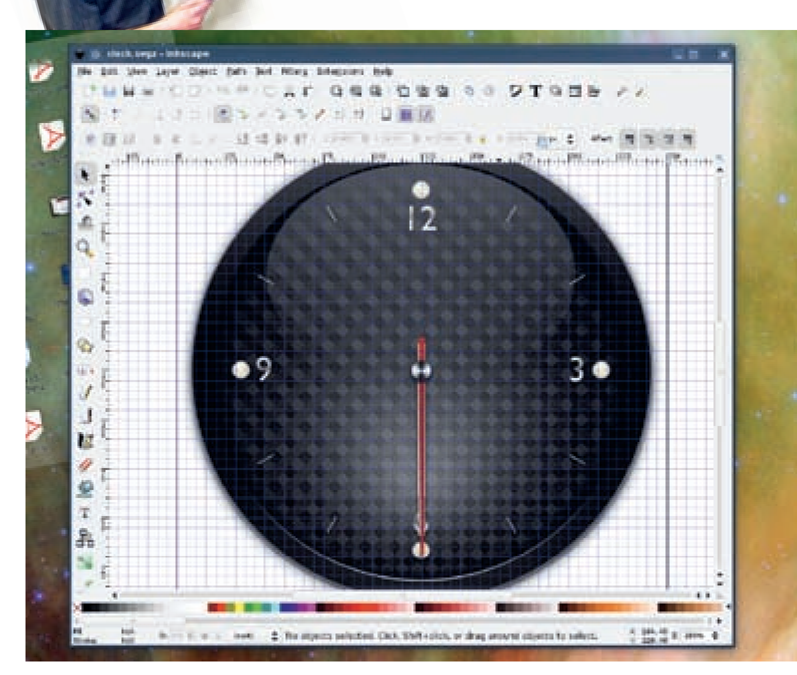
Для каждой работы найдется своя свободная утилита, будь то обрезка изображения для вставки его в документ, создание дизайна сайта или построение диаграммы. Для графического дизайна в мире свободного ПО едва ли не больше возможностей, чем для любых других типов задач. Существуют программы, подходящие любому, от энтузиаста-фотографа до копи-центров и небольшого издательства. *OpenOffice.org* даже предоставляет собственную утилиту для рисования и редактирования изображений.

При этом в мире профессионального дизайна реально есть только одна программа, а именно *Photoshop* от Adobe. Это приложение-гигант, который, кажется, обрастает все новыми и новыми функциями вот уже десяток лет, и в итоге ему сложно обучиться и с него сложно уйти. Если вы ищете альтернативу от свободного ПО, правдивым ответом будет, что такой вам не найти. Однако если ваши потребности в редактировании более скромны, то замена есть, и ее зовут *GIMP*.

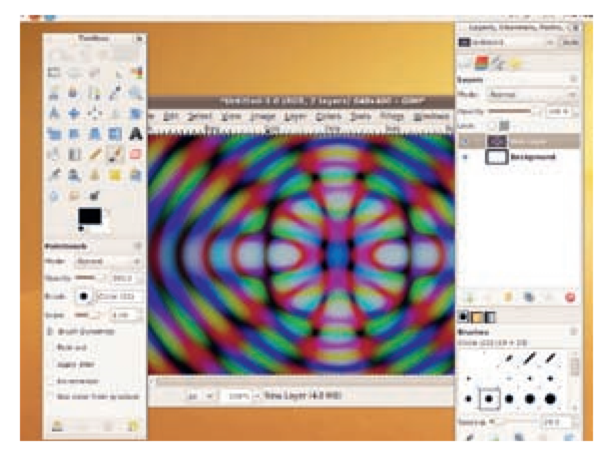
> «Inkscape в основном эквивалентен Illustrator от Adobe.»

Видимо, о *GIMP* все уже слышали – из-за дурацкого названия и довольно загадочного пользовательского интерфейса, который как будто унаследован от одной из старых версий *Photoshop*. Но если отбросить эти недостатки – и довольно крутую кривую обучения – останется сравнительно мощное приложение; лишь копните поглубже. Только не ждите, что простое редактирование удастся безболезненно.

Подобно *Photoshop*, *GIMP* использует слои, чтобы хранить разные части изображения независимо друг от друга. Каждый слой можно фильтровать, изменять, искажать и преобразовывать, и разные режимы слоев могут использоваться для определения, как цвет одного слоя будет влиять на другой. Если вы не пользовались *Photoshop*, вам будет трудно не запутаться, но в основном *GIMP* служит для построения сложных изображений из простых частей, применения эффектов и экспорта всего проекта в один слой.



› Многие значки и виджеты, используемые на рабочем столе Linux, поставляются как изображения SVG, легко редактируемые в *Inkscape*.



› PSD из *Photoshop*'а вы сумеете в *GIMP* открыть, отредактировать и послать обратно в том же формате.

Фотонный окрас

GIMP – это еще и мощное средство для импорта изображений *Photoshop*, имеющих расширение PSD. При этом сохранится большая часть информации о слоях – можно отредактировать файл и послать результат обратно в том же формате. Совместимость не абсолютная, однако если вы свяжетесь с тем, кто шлет вам файл PSD, и попросите визуализировать эффекты на его слоях, то больших проблем не будет.

Для редактирования фотографий у *GIMP* есть широкий диапазон инструментов выделения; для автоматического выбора частей изображения в них можно использовать порог близости цвета, а также комбинировать разные типы выделения. Можно производить обычную корректировку яркости и контраста, делать более сложные подстройки, используя гистограммы кривых цвета или меню, переполненное различными фильтрами и эффектами.

Единственная настоящая проблема *GIMP* – то, что он не имеет родной поддержки цветовой схемы CMYK, хлеба с маслом для профессионалов. Это ограничение можно обойти, установив модуль расширения, создающий цветоделение вручную, но до встроенной поддержки, о которой конечный пользователь даже не задумывается, пока далеко.

Krita

Если вы ищете чего-нибудь попроще в использовании, чем *GIMP*, придется смириться с меньшей мощью. Ваш выбор должен пасть на *Krita*, приложение KDE.

Krita – часть пакета программ *KOffice*, но ее можно установить через менеджер пакетов как отдельное приложение. Как и в *GIMP*, здесь используются слои и набор диалогов, чтобы на экране уместилось как можно больше функций. Есть даже поддержка CMYK. Но главное отличие в том, что по умолчанию *Krita* использует одно окно, не захламляя экран плавающими диалогами, благодаря чему она немного проще в использовании и понимании.

GIMP – мощное решение, и во многих случаях заменяет своего дорогостоящего коллегу. Еще одно его преимущество – наличие версии для Windows, что позволяет вначале запускать его на пару с *Photoshop*; но мы не можем от чистого сердца рекомендовать его взамен *Photoshop*, пока не материализуется, наконец, обещанный новый интерфейс (он запланирован на следующий релиз, версию 2.8), и он станет проще в использовании.

Inkscape

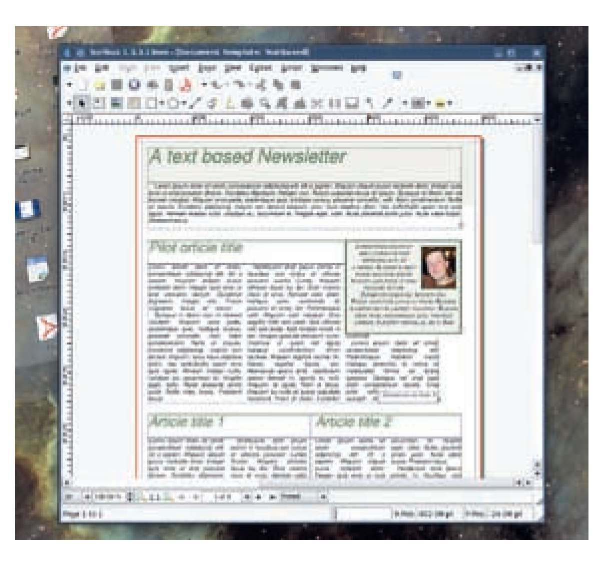
Если вам нужен редактор не растровой, а векторной графики, то вам повезло: на свете есть программа *Inkscape*, которая делает почти все, что нужно, не будучи сложной и непонятной (www.inkscape.org). *Inkscape* пригодится для создания масштабируемых диаграмм, иллюстраций, живописи и текстовых эффектов, и эти творения экспортируются в промышленный стандарт SVG или в PDF.

Inkscape в основном эквивалентен программе *Illustrator* от Adobe, и содержит многие ее функции. Рисунок создается как набор контуров – либо прямых линий, либо гладких кривых, а их стыки отмечают переход к следующему сегменту. Эти формы можно закрашивать и накладывать друг на друга. Палитра слоев, почти такая же, как в *Photoshop* и *GIMP*, хранит независимые фрагменты цельной картины, с которыми можно работать.

Иногда *Inkscape* работает даже лучше, чем *Illustrator*. Например, его родной формат – SVG, и загружать и сохранять файлы этого формата в *Inkscape* проще, а результат – лучше. Некоторыми инструментами, такими как заливка краской, градиент и черчение, пользоваться легче, а для большинства функций есть быстрое сочетание клавиш. В плане промышленного применения он, может, и не конкурент, однако *Inkscape* более чем способен создавать поразительные вещи.

Он также идеален для печати: ведь масштабируемая природа изображений значит, что нет ограничений на разрешение вывода, будь то домашний принтер или коммерческий печатный станок. Он используется подавляющим числом художников открытого ПО при создании всех тех пластичных иконок, имеющихся сейчас в любом программном проекте. Возможно, вы найдете *Inkscape* проще в использовании, чем редактор изображений в *OpenOffice.org*; есть стабильные версии, работающий под OS X и Windows.

Если вам когда-либо приходилось с должным профессионализмом сочетать текст и графику, вы в курсе, что настольные издательские программы малочисленны и сильно разнятся. Однако



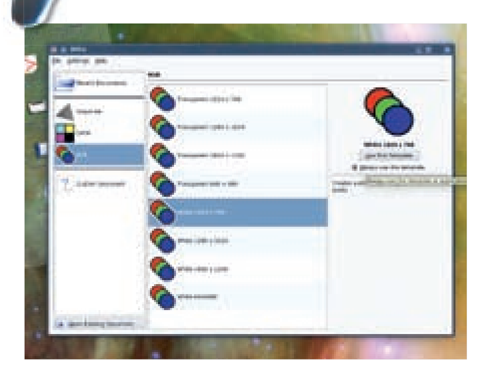
› Если вы пишете информационный бюллетень, рискните отказаться от текстового процессора и дать шанс *Scribus*.

в мире открытого ПО *Scribus* вырос так, что стал стандартом. Это программа, предназначенная для объединения текста и изображений в нечто красиво смотрящееся на бумаге, будь то листовка китайского ресторана, рекламный проспект местного магазина или даже периодическое издание. Как и у *GIMP*, у него нет поддержки привычного рабочего процесса и зрелости коммерческих конкурентов, однако он быстро растет, и для большинства сможет предложить более чем достаточно для решения основных задач.

Scribus

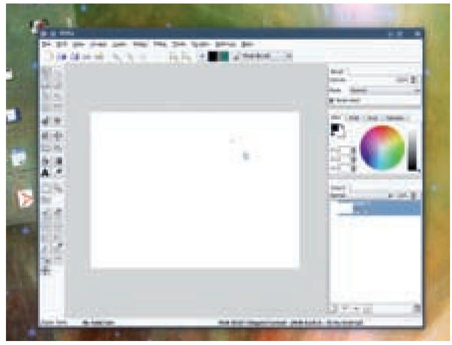
Из-за чрезмерно высокой цены проприетарных издательских пакетов, эти продукты обычно рассматриваются как нишевые, и многим приходится создавать свои информационные бюллетени в текстовом процессоре. Однако если вы не взлетали выше *Word* или *Writer*, *Scribus* сильно улучшит качество ваших изделий (www.scribus.net). А поскольку он свободный, то вы ничего не потеряете, если его испробуете. Существует несколько полезных шаблонов, пригодных для взятия за основу нового проекта в *Scribus*, и в большинстве стандартных ситуаций их достаточно. На www.scribus.net таких целые залежи, в том числе огромный список шаблонов для обложек CD, что еще больше расширяет область применения этой программы и поможет вам и вашим коллегам вырваться из клещей коммерческих программ. LXF

Шаг за шагом: Начало работы в Krita



1 Запустите мастер

Чтобы создать рисунок с нуля, выберите в качестве разрешения 1024×768, а фон возьмите белым. Задайте эти значения по умолчанию.



2 Увеличение вида

Вид, использующийся по умолчанию, слишком мал. Либо воспользуйтесь инструментом «лупа», либо приблизьте его колесиком мыши.



3 Редактируем

Можно копировать/вставлять изображения, брать инструменты рисования и менять свойства слоя, быстро и просто создавая свое творение.



Каждому облаку – коричневую каемку

Сорен Хансен – человек, подтолкнувший Ubuntu к облачным вычислениям. Мы расспросили его, как Canonical чувствует себя в роли новатора...

LINUX FORMAT Интервью

Если, по-вашему, виртуализацию трудно заставить работать на вашем настольном ПК, задумайтесь о деятельности человека, который все это программирует: ведь это совсем иной уровень сложности. Сорен Хансен [Søren Hansen] встроил *KVM* в Ubuntu, но не почил на лаврах: отвечая в Canonical за виртуализацию, он занимается интеграцией в Ubuntu Server системы *Eucalyptus* – это новая инфраструктура облачных вычислений, позволяющая любому сисадмину воссоздать EC2 Elastic Compute Cloud от Amazon.

Мы вытащили Сорена из его уютной норки в чисто американскую забегаловку, чтобы задать вопросы о *Xen*, Ubuntu Server, Java и многом другом...

Linux Format: Что изменилось в вашей работе с выходом на сцену *Eucalyptus*?

Сорен Хансен: В облачных вычислениях заинтересованы все; это действительно вещь крутая. И теперь они свалились на меня. Так что моя работа изменилась весьма существенно! Мы начали присматриваться к *Eucalyptus* [EUCALYPTUS расшифровывается как Elastic Utility Computing Architecture Linking Your Programs To Useful Systems и представляет собой открытый проект, реализующий облачные вычисления в пределах центра управления данными; мы писали про него в LXF129, – *прим. пер.*], потому что знали: нам нужно решение для облачных вычислений. Мы рассматривали все имеющиеся варианты и их применение. И оказалось, что нечто вроде *Eucalyptus* Марку и хотелось, из-за его совместимости с EC2. Облачные вычисления очень расплывчаты – как, э-э, облака.

LXF: Как вы думаете, станут ли облачные вычисления основной темой Ubuntu Server Edition?

СХ: Ну, этого мы не планировали, но сервер требует огромных ресурсов, и мы видим в облачных вычислениях огромный потенциал. Мы нанимаем людей специально для работы с ними. Так что это наш основной упор, и мы надеемся, что именно благодаря им сможем всех опередить. Web-сервер и *MySQL* предлагают все. У любого поставщика Linux имеются аналогичные вещи. Очень сложно уйти в отрыв, пред-



ПРО JAVA

«Миру Java свойственна совершенно иная культура.»

лагая то же, что предлагают и все остальные – здесь можно конкурировать разве что по контрактам на поддержку. А вот облачные вычисления… Пожалуй, в этой области мы лидируем. Red Hat постоянно уверяет, что они тоже реализуют облачные вычисления, но мы пока этого не видим.

LXF: Раз вы — новаторы в этой области, должно быть, начало было не самым легким?

CX: Мы затратили очень много времени на создание первичной версии. Когда мы решили сотрудничать с [*Eucalyptus*] и встраивать его в Ubuntu, нам пришлось работать по очень жесткому графику. Они там не привыкли выпускать ПО, которое не полностью завершено, не прошло через оценку качества и прочее. А мы так работать не могли: нужно было укладываться в сроки завершения всяких альфа- и бета-версий. Раз они хотели, чтобы их проект вошел в Ubuntu, они должны были дать нам доступ к своему коду как можно раньше. И мы получили доступ к их ветви разработки, которая не была открыта для публики.

LXF: Что было самым сложным, когда вы пытались заставить все это работать?

CX: Была масса технических сложностей, не последней из которых оказалась Java – с ней мы работать не привыкли. Я не уверен, что какой-либо из ведущих дистрибутивов Linux работает с Java, и включают ли они Java вообще! Мы потратили жутко много времени, пытаясь пристроить все по местам. Миру Java свойственна совершенно иная культура: у них существует тенденция просто свалить все в огромную кучу, поэтому вместе с *Eucalyptus* вы получаете и встроенную Java-реализацию web-сервисов *Apache*.

Мы так не работаем: в общем случае, если бы мы приняли этот подход и обнаружили проблему с безопасностью, пришлось бы прошерстить все пакеты, использующие проблемный код. А вот если обнаружится проблема в C-библиотеке, мы просто исправим ее, и все будет прекрасно. Поэтому пришлось затратить немало усилий на разделение компонентов, чтобы использовать их как независимые пакеты, а не иметь дело с кодом в виде огромной неупорядоченной кучи.

LXF: Пришлось ли вам убеждать команду *Eucalyptus* в необходимости встраивания совместимости с *KVM*?

CX: Это было настолько просто, что я бы не употреблял слова «убеждать». Они понимали, что на неком этапе это станет неизбежно, а когда осознали, что портят нам песню, потому что *KVM* – наш инструмент виртуализации, вопрос был решен. К тому же они уже использовали *libvirt*, так что в любом случае предвидели нечто подобное.

LXF: А почему вы не взяли *Xen*?

CX: Если коротко, здесь все упиралось в удобство сопровождения. Когда мы принимали решения, в ядре не было и намека на *Xen*, поэтому для поддержки и сопровождения потребовались бы масштабные правки. А система *KVM*, по всему, находилась на правильном пути и имела намного более простую и здравую архитектуру – это мое личное мнение.

LXF: Выбор KVM — это было ваших рук дело?

CX: Нет, по крайней мере, тогда. Я возглавил команду виртуализации случайно. На саммите разработчиков Ubuntu в Бостоне мы обсуждали возможности виртуализации и пришли к выводу, что это перспективное направление. Так что на одном из заседаний мы обсудили различные варианты и составили таблицу достоинств и недостатков каждого из них. Пару месяцев назад я еще раз глянул на нее – мы изрядно заблуждались, но в итоге приняли правильное решение! Но оно было не единоличным.

Сразу же после саммита разработчиков мы решили, что нам нужен сотрудник по виртуализации, и дали объявление о вакансии. Но пока мы не нашли человека, кому-то надо было поручить эту работу, и тут подвернулся я. Через три или четыре месяца мы отчаялись найти подходящего кандидата, а мой босс счел, что моя деятельность на этом посту не совсем уж чудовищна. Вот я и стал ведущим по виртуализации.

LXF: Вы не считаете, что Ubuntu вступает в игру с небольшим опозданием?

CX: Да, мы немного запоздали с виртуализацией в целом, зато первыми начали работать с *KVM*. Поэтому, хоть мы и явились последними, но все равно захватили лидерство: сейчас все переходят на *KVM*. Перед релизом Hardy я недосыпал добрый месяц или около того, потому что именно тогда мы начали поставлять *KVM*. Эта была большая работа, но, я думаю, игра стоила свеч – мы вышли на очень хорошую позицию. А недавно я обнаружил, что сообщество *KVM* истинно благодарно нам за вывод *KVM* в свет: это дает им хороший аргумент, позволяющий убеждать людей в необходимости перехода на *KVM* и оправдывать затраты на этот переход перед своими работодателями.

LXF: Что произойдет, когда миллионы пользователей Ubuntu получат *Eucalyptus*?

CX: Я не уверен, что хочу об этом думать… В идеальном мире следовало бы провести куда более масштабное тестирование. Но по-моему, это вряд ли составит реальную проблему. Большинству людей он просто ни к чему.

LXF: Он выглядит весьма специфичным…

CX: Так и есть! Это – очень нишевый продукт. Но те, кто собирается им пользоваться, будут проводить масштабные развертывания – сотни и тысячи узлов. А это и есть та самая группа пользователей, которая готова платить за контракты по техподдержке! LXF



Linux-аудио в деталях

Грэм Моррисон докопался до центра ядра Linux, разбираясь, почему звук несозвучен здравому смыслу.

С аудио в Linux имеется проблема – и речь не о том, что данная система не всегда работает: беда в ее чрезмерной сложности. Попробуйте-ка набросать на бумаге схему взаимосвязей между всеми технологиями Linux, участвующими в превращении аудиофайла в мелодию: она мигом уподобится вороху макарон! И это провал: ведь у обработки звука нет причин отличаться от других технологий. Информация входит в ваш Linux-компьютер, преобразуется и поступает на выход.

Если бы мы рисовали схему модели OSI – она описывает инфраструктуру, обеспечивающую взаимодействие вашего компьютера с другими машинами сети – то увидели бы четкие уровни, ответственные каждый за свою группу процессов и функций и практически не перекрывающиеся. На седьмом, прикладном уровне процессы пользователей не занимаются обработкой электрических импульсов от потока битов: это задача процессов первого уровня, физического.

> «Это скорее модель земной коры, чем сетевая модель.»

А вот в Linux-аудио такое бывает сплошь и рядом. Здесь даже нет четкого определения нижнего уровня, и между аппаратурой и ядром суетится несколько независимых друг от друга аудиотехнологий. Звуковая архитектура Linux – это скорее модель земной коры, чем сети: нижние уровни иногда извергаются на поверхность, вызывая оторопь, а верхние сползают и обнажают «пласты» нижележащих технологий, которым полагается быть скрытыми. Например, протокол Open Sound ранее использовался на уровне ядра для прямого диалога с аппаратурой, а теперь это слой совместимости поверх *ALSA*. *ALSA* же представляет собой стек уровня ядра плюс API высокого уровня для программистов, смешавший драйверы и свойства аппаратных компонентов с функцией воспроизведения объемного звука или кодека MP3. Большинство дистрибутивов еще и громоздят над этим *PulseAudio* и *GStreamer*, и получается очаг нестабильности похлеще разлома Сан-Андреас в Калифорнии [эпицентр множества землетрясений, – *прим. пер.*].



ALSA

ВХОДЫ: *PulseAudio*, Jack, *GStreamer*, *Xine*, *SDL*, *ESD*
ВЫХОДЫ: аппаратура, OSS

Как говорится, танцевать начнем от печки. Для современного Linux-аудио этой «печкой» является *ALSA* (Advanced Linux Sound Architecture, Продвинутая звуковая архитектура Linux). *ALSA* связана с ядром Linux и обеспечивает работу со звуком для всех компонентов системы. Но *ALSA* – не просто драйвер ядра: она служит программным микшером, эмулирует совместимость для других аудиослоев, создает API для программистов и работает с такой низкой и стабильной латентностью, что способна конкурировать со своими эквивалентами ASIO (Audio Stream Input/Output) и CoreAudio из Windows и Mac OS X.

ALSA создавалась на замену OSS (Open Sound System). Однако драйвер OSS все же не умер: в *ALSA* предусмотрена совместимость со старыми приложениями, поддерживающими только OSS. Проще всего считать *ALSA* уровнем драйверов устройств в системе Linux-аудио. Вашей звуковой аппаратуре нужен соответствующий модуль ядра, с префиксом **snd_**, а тот надо загрузить и запустить. И чтобы услышать хоть какой-нибудь звук, не обойтись без драйвера ядра из *ALSA*. Вот почему ваш ноутбук так долго оставался «немым» – пока у кого-то не дошли руки написать для него звуковой драйвер. К счастью, в наше время большинство дистрибутивов Linux настраивают ваши устройства и модули автоматически.

ALSA переводит функциональность звуковой аппаратуры в программный API, которым прочие компоненты системы пользуются для манипуляции звуком. Эта архитектура создавалась для устранения недостатков OSS (и большинства аудиодрайверов того времени), а самым заметным из них была невозможность работы с аудиоустройством сразу нескольких приложений. Задача программного компонента *ALSA* – управление запросами к звуковым устройствам, с учетом возможностей этих последних.

Например, если вы хотите играть в какую-нибудь игру еще и под музыку из *Amarok*, *ALSA* нужно принимать оба потока аудио и либо микшировать их программно, либо отсылать их на аппаратный микшер звуковой карты. *ALSA* умеет управлять работой до 8 устройств одновременно, а иногда и обращается к аппаратным функциям MIDI, хотя это зависит от звукового драйвера вашего оборудования и по мере роста мощности компьютеров теряет актуальность.

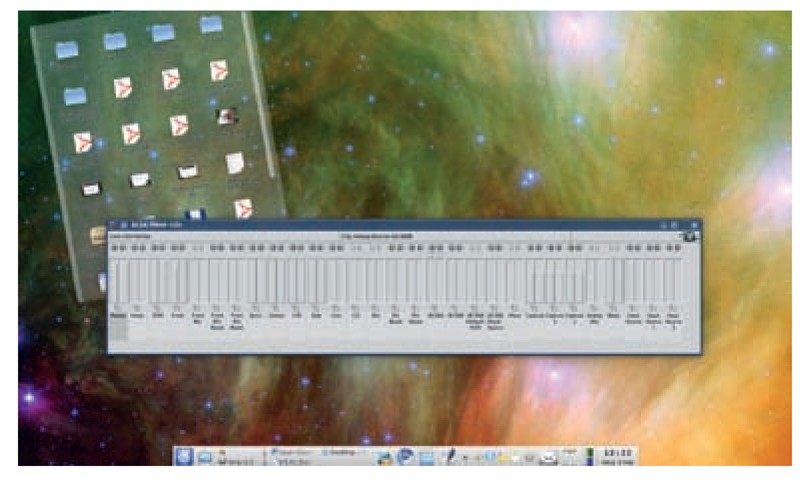
В отличие от типичного модуля ядра/драйвера устройства, *ALSA* предоставляет возможность своей настройки. Здесь-то и проявляется сложность аудиосистемы Linux. Стоит лишь создать собственный файл настройки, и в конфигурации *ALSA* можно сменить практически все: от способа микширования потоков аудио и выбора аудиовыходов до частоты дискретизации, разрядности отсчетов звука и эффектов реального времени.

Относительная «прозрачность», эффективность и гибкость сделали *ALSA* стандартом де-факто; ее слой должна пройти любая другая инфраструктура аудио, чтобы добраться до звукового оборудования.

PulseAudio

ВХОДЫ: *GStreamer*, *Xine*, *ALSA*
ВЫХОДЫ: *ALSA*, Jack, *ESD*, OSS

Если вы решили, что *ALSA* упростила ситуация с Linux-аудио, погодите радоваться. Конечно, *ALSA* в основном охватывает всю «матчасть» вашего компьютера по вводу/выводу звука, но вы должны прорваться через другой уровень сложности – вотчину *PulseAudio*. Данный компонент – попытка перебросить мост между аппаратурой и ПО, локальными и удаленными компьютерами, а также содержимым аудиопотоков. *PulseAudio* играет при передаче аудио через сеть такую же роль, как *ALSA* при работе с несколькими звуковыми картами. Благодаря своей гибкости, *PulseAudio* стал стандартом во многих дистрибутивах Linux.



› **Чудовищно сложный вид по умолчанию из микшера *ALSA* для типового звукового устройства наглядно иллюстрирует проблему Linux-аудио.**

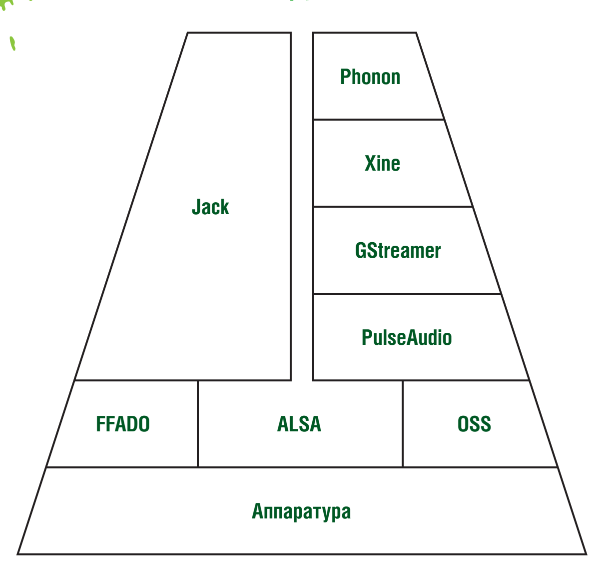
Как и в *ALSA*, эта гибкость достигается за счет усложнения, что усугубляется выходом *PulseAudio* на пользователей. То есть рядовые пользователи с большей вероятностью запутаются в его паутине. Большинство дистрибутивов держат настройку *PulseAudio* за пределами досягаемости. Например, в последнем релизе Ubuntu вы и не заметите, что *PulseAudio* установлен. Щелкнув мышью по апплету микшера, чтобы настроить громкость звуковой карты, вы увидите панель *ALSA*, но по сути перед вами виртуальное устройство – вывод *ALSA* передается на *PulseAudio* и обратно на *ALSA*.

На первый взгляд кажется, что *PulseAudio* не внес в Linux-аудио ничего нового, и многие воспринимают его враждебно. *PulseAudio* не упрощает уже имеющейся системы, не делает ее более надежной, но зато добавляет несколько важных функций. И это универсальный слой для всех приложений, работающих с Linux-аудио, независимо от их индивидуальных особенностей или от имеющихся аппаратных средств.

Если бы все приложения применяли *PulseAudio*, ситуация бы упростилась. Разработчики могли бы не беспокоиться о сложностях других систем, потому что *PulseAudio* обеспечивает кросс-платформенную совместимость. Но это же является основной причиной, объясняющей, почему существует так много разнообразных решений проблемы аудио. *PulseAudio* может работать на разных платформах, включая POSIX и Microsoft Windows. А значит, приложение, созданное на основе *PulseAudio*, а не *ALSA*, будет легко перенести на другие платформы.

> «В конфигурации ALSA можно сменить практически все.»

Но *ALSA* и *PulseAudio* образуют некий симбиоз: второй нуждается в первом, чтобы «выжить» в Linux. *PulseAudio* настраивается как виртуальное устройство, соединенное с *ALSA* подобно аппаратному устройству. В этом *PulseAudio* походит на Jack: он тоже сидит между *ALSA* и рабочим столом и прозрачным образом перебрасывает между ними данные туда и обратно. *PulseAudio* применяет собственную терминологию: например, конечные пункты назначения при пересылке аудиоданных называются «стоками» [sinks]. Ими могут быть другие компьютеры сети или аудиовыходы вашей звуковой карты (благодаря виртуализации *ALSA*). Компоненты *PulseAudio*, наполняющие эти «стоки» информацией, называются «источниками» [sources]. Обычно это приложения, генерирующие аудиовывод, аудиовходы вашей звуковой карты или сетевой аудиопоток, отправляемый в сеть другим компьютером с *PulseAudio*.



› Упрощенная схема типичных для Linux уровней аудио. Чем ниже уровень, тем ближе он к аппаратным средствам.

В отличие от Jack, приложения не отвечают за добавление и удаление источников напрямую, и вы получаете более тонкую степень управления потоками. Например, микшером *PulseAudio* можно выставлять относительные громкости всех приложений, передающих звук через *PulseAudio*, независимо от того, есть у них свои регуляторы громкости или нет. Это отличный способ заглушить чересчур шумные сайты.

GStreamer

ВХОДЫ: *Phonon*

ВЫХОДЫ: *ALSA*, *PhaseAudio*, Jack, *ESD*

GStreamer вносит в Linux-аудио еще большую путаницу – потому что на первый взгляд, как и *PulseAudio*, не добавляет в эту смесь ничего нового. Это еще одна инфраструктура мультимедиа, до *PulseAudio* довольно популярная среди разработчиков, особенно для рабочего стола Gnome. Она предоставляет один из немногих простых способов установки и использования в Linux проприетарных кодеков. Кроме того, эта звуковая инфраструктура принята у разработчиков *GTK*, и можно даже найти версию для обработки аудио на коммуникаторах Palm Pre.

На схеме слоев *GStreamer* располагается поверх *PulseAudio* (который используется им для вывода звука в большинстве дистрибутивов), но ниже прикладного уровня. Уникальность *GStreamer* в том, что этот компонент не предназначен исключительно для работы с аудио и поддерживает различные форматы потоковой мультимедиа-информации, включая и видео – посредством модулей расширения.

Например, воспроизведение MP3 добавляется в систему за счет загрузки и установки дополнительного кодека, добавляемого как расширение *GStreamer*. Такими расширениями являются коммерческий MP3-декодер *Fluendo*, один из немногих официально лицензированных для Linux, а также проприетарные кодеки MPEG-2, H.264 и MPEG.

› С помощью Jack можно вручную соединять аудиовыходы одних приложений с аудиовходами других — прямо как с устройствами в настоящей студии звукозаписи.

Jack

ВХОДЫ: *PulseAudio*, *GStreamer*, *ALSA*

ВЫХОДЫ: OSS, FFADO, *ALSA*

При всех преимуществах открытых конфигураций наподобие *PulseAudio*, они перекидывают звук между приложениями, предполагая, что оно проследует непосредственно на их выходы. Jack – это промежуточный уровень, играющий в мире аудио роль, аналогичную удаленным вызовам процедур [Remote Procedure Calls, RPC] в программировании. Благодаря Jack аудиоприложения можно строить, собирая из компонентов.

Лучший пример использования Jack – виртуальная студия звукозаписи, где одно приложение отвечает за захват аудио, а другое – за его обработку и наложение эффектов перед отправкой результирующего аудиопотока третьему приложению, в котором выполняется сведение и подготовка записи к релизу. Настоящая студия звукозаписи опутана целой паутиной кабелей с разъемами [jack]. Компонент Jack делает то же, но на программном уровне.

Слово Jack – это акроним от 'Jack Audio Connection Kit'. Латентность Jack низка, то есть над аудиопотоком не выполняется никаких лишних операций, способных его затормозить. Но чтобы Jack приносил пользу, нужны аудиоприложения, специально подогнанные к соединениям Jack. Поэтому Jack не может просто заменить *ALSA* или *PulseAudio*, а должен работать поверх другой системы, которая генерирует звук и предоставляет физические входы.

Работая с Jack-совместимыми приложениями, вы можете направлять аудио и подключать входы любым удобным для вас способом. Например, выход *VLC* можно перенаправить на вход *Audacity*, чтобы записать поток в процессе его воспроизведения, или пропустить аудиопоток через *JackRack* – программный процессор сигналов, позволяющий нагромоздить целую гору эффектов в реальном времени, включая эхо (в том числе многократное пещерное) и придание голосу приятного тембра.

Эта гибкость создает фантастические возможности для рабочих станций, занимающихся цифровым звуком. Например, *Ardour* использует Jack для внутренних и внешних соединений, а эффект-процессор *Jamin* может служить одним из звеньев в цепочке процессов Jack. Jack – как бы программный эквивалент подключений в настоящей студии звукозаписи, и его реализация в Linux оказалась столь успешной, что даже перекочевала в Mac OS X.

FFADO

ВХОДЫ: Jack

ВЫХОДЫ: Аудиоаппаратура

В мире профессиональной и полупрофессиональной звукозаписи многие аудиоинтерфейсы связываются с хост-компьютером через порт FireWire. Такой подход обладает массой преимуществ. FireWire – интерфейс скоростной, а устройства могут питаться от шины. Стандарт FireWire существует уже давно, это зрелая и надежная технология. Порты FireWire имеются на многих ноутбуках и настольных компьютерах и не требуют лишней доработки. Устройства FireWire можно взять с собой на выезд для записи с ноутбука, а вернувшись в студию, вновь подключить их к настольному компьютеру.

Однако, в отличие от USB, где предусмотрен специальный стандарт обработки аудио без дополнительных драйверов, аудиоинтерфейсы FireWire имеют индивидуальные драйверы. Сложности протокола FireWire затрудняют взаимодействие с *ALSA*, и для этого необходим отдельный уровень. Изначально данную задачу выполнял проект FreeBOB – ему повезло в том, что многие аудиоустройства FireWire применяли одинаковую аппаратуру.

FFADO – преемник FreeBOB, открывающий платформу драйверов для многих других типов звуковых устройств с интерфейсом FireWire. В конце 2009 года вышла вторая версия FFADO, с поддержкой ряда устройств от таких производителей оборудования, как Alesis, Apogee, ART, CME, Echo, Edirol, Focusrite, m-Audio, Mackie,

Phonic и Terratec. Выбор поддерживаемых устройств в общем случаен; перед покупкой не помешает навести справки. Но многие из этих производителей помогают разработке драйверов, предоставляя свои устройства для использования и тестирования.

Еще одна выигрышная черта FFADO – интеграция ряда функций цифровой обработки сигналов при аппаратном микшировании в драйвер, дополненный графическим микшером для управления балансом различных входов и выходов. В отличие от микшера *ALSA*, управление аудиопотоками идет на оборудовании с нулевой латентностью, что идеально при записи «живого» звука.

По сравнению с остальными аудиоуровнями, FFADO только «перетасовывает» аудиопотоки между Jack и физическим звуковым оборудованием. «Обратного хода» от FFADO к *PulseAudio* или *GStreamer* нет, если только они не запущены вместе с Jack. Поэтому нельзя сделать FFADO основным уровнем аудио для воспроизведения музыки или фильмов – разве что вам по зубам установить все это и запустить Jack. Зато драйвер не обременен поддержкой разнообразных протоколов, поскольку большинство «серьезных» аудиоприложений уже включают в свой состав поддержку Jack по умолчанию; так что FFADO – один из лучших вариантов для использования в условиях студии звукозаписи.

Xine

ВХОДЫ: *Phonon*
ВЫХОДЫ: *PulseAudio*, *ALSA*, ESD

Мы переходим к узкоспециальной геологии Linux-аудио. *Xine* – своего рода «меловые отложения», остаток после «вымывания» прочих слоев. Большинству пользователей слово *Xine* знакомо: так называется очень удачный проигрыватель DVD, включенный во многие дистрибутивы, несмотря на свой почтенный возраст. Чем и обусловлено долголетие *Xine*.

Xine разбит на внутреннюю библиотеку для обработки медиа и интерфейс для общения с пользователем. Первая обязана своей живучестью умению воспроизводить уйму контейнерных форматов, включая AVI, Matroska и Ogg, а также десятки других, которые могут содержаться в этих контейнерах, в том числе AAC, Flac, MP3, Vorbis и WMA – благодаря привлечению ряда других библиотек. В итоге *Xine* действует как универсальная инфраструктура, дающая разработчикам наилучший охват совместимости без тревог о законности применения проприетарных кодеков.

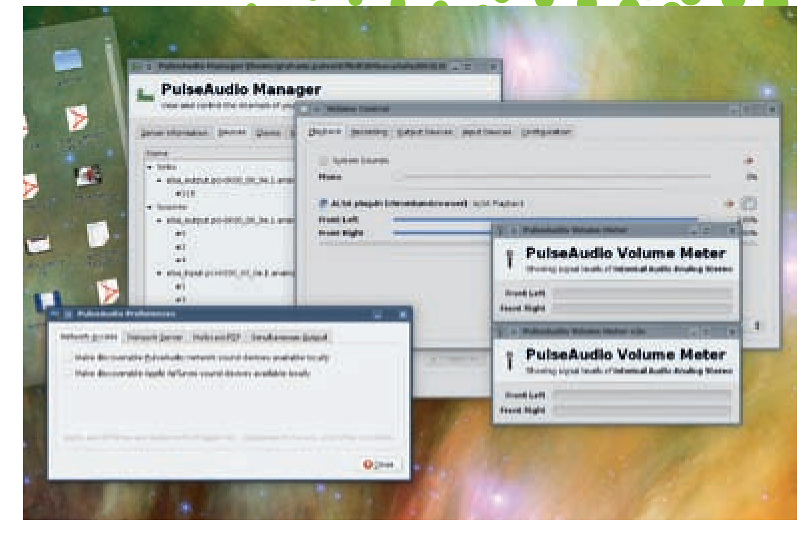
Xine взаимодействует с *ALSA* и *PulseAudio*, передавая им свою выходную информацию, и число приложений, умеющих «общаться» с *Xine* напрямую, по-прежнему велико. Среди них наиболее популярны *Gxine* и *Totem*. Вдобавок *Xine* – движок по умолчанию для *Phonon* в KDE, встречаемый повсюду – от *Amarok* до *Kaffeine*.

Phonon

ВХОДЫ: *Qt* и приложения KDE
ВЫХОДЫ: *GStreamer*, *Xine*

Компонент *Phonon* разрабатывался, чтоб упростить жизнь и разработчикам, и пользователям за счет устранения ряда сложностей системы, и вступил в жизнь как еще один уровень абстракции аудио для приложений KDE 4, но идея оказалась столь хороша, что разработчики *Qt* включили его непосредственно в инструментарий *Qt*, на котором базируется и сама среда KDE.

От этого здорово выиграли разработчики кросс-платформенных приложений: появились шансы создать в Linux проигрыватель музыки с помощью *Qt*, а затем просто перекомпилировать его для Mac OS X или Windows, не задумываясь о применяемом звуковом оборудовании и способе воспроизведения музыки или обработки аудио в целевой ОС. *Qt* и *Phonon* взяли все это на себя, и автоматически передают аудиоданные CoreAudio API в Mac OS X или DirectSound в Windows. В Linux (в отличие от исходной версии *Phonon* в KDE) *Qt*-реализация *Phonon* передает данные на вход *GStreamer* – тот обеспечивает прозрачную поддержку кодеков.



› *PulseAudio* — **мощная технология, но ее часто осмеивают за внесение осложнений в систему Linux-аудио.**

Поддержка *Phonon* потихоньку изымается из инструментария *Qt*. Систему часто критиковали, в основном за чрезмерную упрощенность и отсутствие новизны. Впрочем, вполне вероятно, что поддержка все-таки протянет до конца жизненного цикла KDE 4.

Прочая компания

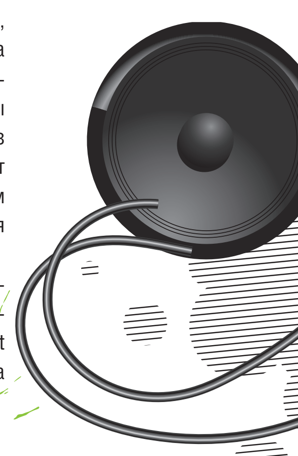
Существует еще немало технологий работы со звуком, в том числе *ESD*, *SDL* и *PortAudio*. Звуковой сервер для оконного менеджера *Enlightenment* (*Enlighted Sound Daemon*, *ESD*) долгое время применялся по умолчанию в графической оболочке Gnome. Потом Gnome перебрался на *libcanberra* (эта библиотека может общаться с *ALSA*, *GStreamer*, OSS и *PulseAudio*), а поддержку *ESD* в апреле 2009 прекратили. В KDE также имеется *Arts*, аналог *ESD*, хотя и менее популярный: он, пожалуй, создавал больше проблем, чем решал. Сейчас многие перешли на KDE 4, и об этом уже речи нет.

> «SDL поддерживает массу функций, и это зрелая технология.»

В качестве компонента аудиовывода библиотеки *SDL* также процветает *SDL*, применяемый в сотнях кросс-платформенных игр. *SDL* поддерживает массу функций, и это зрелая и стабильная технология. Еще одна кросс-платформенная аудиобиблиотека – *PortAudio*; к перечню доступных ОС она добавляет SGI, Unix и BeOS. Самый популярный потребитель *PortAudio* – музыкальный редактор *Audacity*, чем объясняется некоторая непредсказуемость его звукового вывода и качество поддержки Jack.

Наконец, есть еще OSS, Open Sound System. Начиная с версии ядра 2.4, система OSS более не является ключевым компонентом Linux-аудио, но это не пошатнуло ее позиций. Отчасти это связано с тем, что от OSS зависит много старых приложений, а также с тем, что, в отличие от *ALSA*, OSS работает не только с Linux, но и с другими ОС: есть даже версия для FreeBSD. Для 1992 года это была отличная система, но взамен ей практически везде рекомендуют *ALSA*. Система OSS определяла работу аудиосистемы в Linux, и, в частности, способ доступа к аудиоустройствам через дерево ioctl, как, например, в случае с **/dev/dsp**. *ALSA* реализует уровень совместимости с OSS, чтобы дать возможность старым приложениям работать с OSS без необходимости отказываться от текущей версии стандарта *ALSA*.

Проект OSS занимался экспериментами и с открытым кодом, и с проприетарными технологиями. На коммерческой основе проект до сих пор активно поддерживается компанией 4 Front Technologies. Последняя сборка OSS, OSS v. 4.2 Build 2002, вышла в ноябре 2009 года. LXF



Dzen: апплеты на заказ

Принято считать, что современный рабочий стол немыслим без виджетов. **Павел Воронцов** научит запрограммировать их самостоятельно – на ваш вкус.



Наш эксперт

Павел Воронцов предпочитает великолепию KDE и Plasma рациональную гармонию Perl и Dzen.

В той или иной степени, интерес к настройке рабочей среды есть у каждого пользователя. Одних не волнует внешний вид: удобство и быстродействие – их главный приоритет, а остальное – баловство. Другие готовы нарядить свой рабочий стол, как новогоднюю ёлку, и вопрос быстродействия их не волнует. Лично я придерживаюсь точки зрения, что в рабочем окружении, как и в человеке, всё должно быть прекрасно: и быстродействие, и удобство, и внешний вид. Если вы считаете так же, вам непременно следует взглянуть на *Dzen*, тем более что в последнее время он становится всё более популярным.

Что же такое *Dzen*? Вообще-то это маленькая программа, весьма ограниченная в возможностях. Она предназначена для создания информационных панелей, всплывающих напоминаний и меню. Основная задача – графический вывод текста, который подаётся на её стандартный ввод. *Dzen* удобно использовать в скриптах, для передачи результата их работы в графическом виде на рабочий стол. Обычно она используется в паре со сверхлегким оконным менеджером *xmonad*, но ничто не мешает применить её и в любом другом окружении. То есть, *Dzen* – это не программа, которая может всё, а лишь инструмент, который поможет вам вывести нужную информацию в нужное время и в нужном месте. Всё остальное придётся делать самому. Этим мы сейчас и займемся.

Что-то с памятью моей...

Для примера, напишем монитор памяти. Он есть во многих программах, но чаще всего в них больше функций, чем реально необходимо, они занимают больше памяти (пардон за каламбур) и системных ресурсов, чем нужно, и кроме того, вывод результата мониторинга осуществляется только в том виде, в котором задумал автор. Количество свободных килобайт памяти меня не интересует – меня вообще не интересует состояние памяти, когда её много. Мне нужен маленький индикатор, который будет мигать, если памяти мало, и постоянно гореть, если её почти не осталось. Найти программу, которая удовлетворит этим требованиям, практически невозможно, поэтому мы напишем её сами, используя *Dzen* (файл **mem.pl** на диске):

```
#!/usr/bin/perl
$na = "^fg(#505050)"; $active = "^fg(#4F95DE)"; $critical = "^fg(red)";
open (DZEN, "| dzen2 -fn -*-fixed-*-r-*-*-12-*-*-*-*-*-*-* -bg '#303030' -x 40 -y 0 -w 40 -ta c") || die "dzen2-ошибка\n";
select (DZEN); $| =1;
for (;;) {
$mem = `free`;
if ($mem =~ /Mem:\s+(\d+).*\n-\D+(\d+).*\nSwap:\s+(\d+)\s+(\d+)/) {
  $totalm=$1;$usem=$2;$totals=$3;$uses=$4;
  if ($totalm != 0) {$usemp = int($usem*100/$totalm);} else {$usemp = 0;}
  if ($totals != 0) {$usesp = int($uses*100/$totals);} else {$usesp = 0;}
  if ($usemp < 80) {$color = $na;}
  elsif ($usemp > 95 and $usesp > 90) {$color = $critical;}
  elsif ($i == 0) {$color = $na; $i = 1;}
  elsif ($i == 1) {$color = $active; $i = 0;}
}
print " $color\MEM\n";
sleep 1;
}
```

Первая строка указывает на используемый интерпретатор. В нашем случае это **/usr/bin/perl**. Разумеется, Perl должен быть установлен в системе.

Вторая присваивает переменным значения цвета: активный, неактивный и критический. Обратите внимание на формат "^fg(color)". Это код форматирования *Dzen*, означающий, что надо поменять цвет всех последующих символов на заданный. (Соответственно, код ^bg(color) меняет цвет фона). Префикс # предваряет задание шестнадцатеричного значения цвета, как в HTML.

Строка 3 запускает dzen2 и связывает его стандартный ввод с дескриптором файла DZEN, который будет использоваться для передачи форматированного текста. Обратите внимание на то, что dzen2 запускается с определёнными параметрами:

» **-fn** – имя шрифта в формате X Window (для выбора воспользуйтесь *xfontsel*).

» **-bg** – стандартный цвет фона.

» **-fg** – цвет шрифта.
» **-x** и **-y** – абсолютные координаты расположения окна в пикселях относительно левого верхнего угла экрана (у нас 40, 0).
» **-w** – длина строки в пикселях.
» **-h** – высота, по умолчанию равна высоте шрифта + 2 пикселя.
» **-ta** – выравнивание заголовка. Если строка одна, как в данном примере, это и есть заголовок; если строк больше, то заголовком является самая верхняя строка, а остальные принадлежат к так называемому ведомому окну. Варианты выравнивания: **l** – по левому краю, **r** – по правому и **c** – по центру.

Выражение $| =1 отключает буферизацию текста, чтобы результат сразу выдавался на стандартный вывод. Это нюанс, который нужно учитывать при обмене данными между программами.

Далее идёт бесконечный цикл (for (;;) {…}) с задержкой в одну секунду, выполняющий код, который мы не будем рассматривать детально. Он анализирует вывод команды free, рассчитывает процент занятой памяти и, в зависимости от результата, меняет цвет по алгоритму: если занято менее 80 % памяти – неактивный, если больше 95 % физической памяти и 90 % подкачки – критический. Во всех остальных случаях на каждой итерации цикла активный и неактивный цвета чередуются – происходит мигание. Размер занятой памяти берётся с учетом кэша – так объективнее. И, естественно, при каждом проходе цикла результат посылается на стандартный ввод **dzen**. Использование **free** в качестве источника данных упрощает код, но делает его менее эффективным; в качестве упражнения, попробуйте читать напрямую **/proc/memstat** и избавиться от постоянного вызова внешней программы.

> «Dzen может отображать простые графические элементы.»

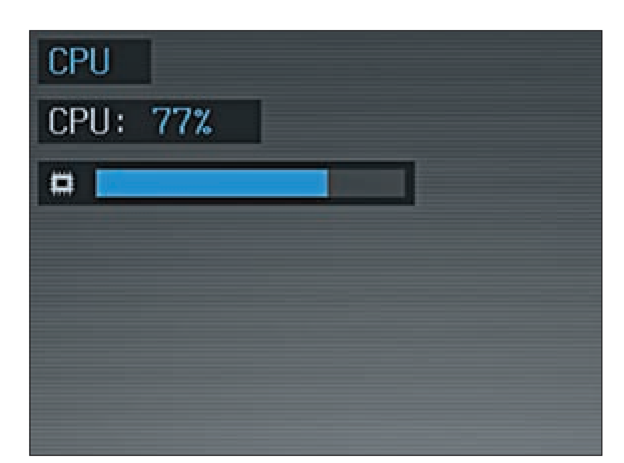
Загрузка процессора

Давайте теперь рассчитаем и выведем на панель текущую загрузку процессора. Для решения этой задачи можно, как и раньше, использовать внешнюю команду (*top*), но мы пойдем другим путем: вычислим нагрузку самостоятельно по данным файла **/proc/stat**. Ниже показана одна из реализаций такого алгоритма (**cpu.pl**):

```
#!/usr/bin/perl
$na = "^fg(#505050)"; $active = "^fg(#4F95DE)"; $critical =
"^fg(red)";
open (DZEN, "| dzen2 -fn -*-fixed-*-r-*-*-12-*-*-*-*-*-*-* -bg
'#303030' -x 0 -y 0 -w 40 -ta c") || die "dzen2 - ошибка\n";
select (DZEN); $| =1;
for (;;) {
$all=$idle=0;
open (CPU,"/proc/stat") || die "stat-ошибка!\n"; $_ = <CPU>;
close(CPU);
@cpu = split(/\s+/); shift (@cpu);
$idle = @cpu[3];
foreach (@cpu) {$all += $_;}
$diffidle = $idle - $preidle; $diffall = $all - $preall;
$diffuse = int((1000*($diffall-$diffidle)/$diffall+5)/10);
if ($diffuse < 70) {$color = $na}
elsif ($diffuse == 100) {$color = $critical}
elsif ($i == 0) {$color = $na; $i =1;}
elsif ($i == 1) {$color = $active; $i =0;}
print "$color\CPU\n";
$preall = $all; $preidle = $idle;
sleep 2
}
```

Если хотите, можете использовать и цифровые значения: например, переменная $diffuse содержит текущую загрузку процессора в процентах. Результат можно вывести так:

```
print "CPU: $active$diffuse%\n";
```



› **Dzen отображает загрузку процессора тремя разными способами.**

Еще интереснее сделать графический индикатор. Да, *Dzen* может отображать графические элементы: простейшие фигуры (прямоугольники, окружности) и растровые изображения в формате XBM и XPM. Делается это посредством специальных кодов форматирования. При помощи геометрических фигур можно строить различные индикаторы: в некоторых случаях их использование более наглядно, чем обычный текст. Чтобы облегчить задачу скриптописцам, вместе с *Dzen* поставляются специальные утилиты (*dbar* и *gdbar*), которые, в зависимости от входных данных (цифровое значение), выводят уже отформатированную строку, описывающую индикатор, которую, в свою очередь можно подать на вход *Dzen*. В некоторых случаях использование этих утилит значительно уменьшает код скрипта. *Dbar* форматирует индикатор, используя текстовые символы, а *gdbar* применяет геометрические фигуры. Вот пример вывода (без дополнительных параметров, шкала – от 0 до 100):

```
$ echo 25 | dbar ; echo 25 | gdbar

25% [======                    ]
25% ^fg(white)^r(20x10)^fg(darkgrey)^r(60x10)^fg()
```

Если говорить о нашем скрипте, то результат можно выводить следующим образом:

```
$bar = `echo $diffuse | gdbar -w 60 -h 8 -bg '#505050' -fg
'#4F95DE' -nonl`;
$bar =~ s/^.*%//;
print "^i(icons/cpu.xbm)$bar\n";
```

Разумеется, самый информативный индикатор – график загрузки по времени. Реализовать его несложно – нужно запоминать текущее значение загрузки в массиве и сдвигать с каждым новым циклом; но правильно отображать его нелегко. Расположение графического элемента определяется только одной горизонтальной координатой, по вертикали он выравнивается по центру, поэтому получить ровный и визуально понятный график в текущей версии *Dzen* (0.8.5) сложно. Но это вопрос времени – в последних SVN-срезах *Dzen* стал гибче в плане форматирования. Появилась возможность более точно выводить графические фигуры и создавать более сложные объекты, такие как графики и вертикальные индикаторы загрузок.

Прогноз погоды

Теперь немного расширим зону наших интересов. Пусть нам нужен прогноз погоды. В принципе, данную информацию можно скачивать с weather.com – так делает большинство программ та- »





кого типа; но нам интересен другой источник, например, gismeteo.ru. В общем случае, можно брать интересующие нас сведения прямо с web-страницы, разбирая её с помощью регулярных выражений, как мы делали в первом примере. Минус такого способа очевиден: структура страниц может поменяться, и, конечно, они содержат очень много избыточной информации, которую нужно как-то фильтровать. К счастью, российский погодный сервер gismeteo.ru предоставляет пользователю XML-файл с подробным прогнозом на сутки вперёд (4 прогноза с шагом по 6 часов). Найти ссылку вы сможете на странице informer.gismeteo.ru, там же расположено и описание формата. Ниже я приведу пример соответствующего скрипта (**meteo.pl**). Так как выводимой информации будет много, я буду использовать *Dzen* в форме раскрывающегося списка, появляющегося при наведении курсора мыши.

```
#!/usr/bin/perl
use LWP::Simple;
use XML::Simple;
$option="^fg(white)";$value="^fg(#4F95DE)";
open (DZEN, "| dzen2 -fn -*-fixed-*-r-*-*-12-*-*-*-*-*-*-* -tw 140 -bg '#303030' -x 80 -y 0 -w 160 -tw 120 -ta c -sa c -l 22 -e 'entertitle=uncollapse;leavetitle=collapse;'") || die "dzen2-ошибка!\n";
select (DZEN); $| = 1;
@month = qw/месяц января февраля марта апреля мая июня июля августа сентября октября ноября декабря/;
@tod = qw/НОЧЬ УТРО ДЕНЬ ВЕЧЕР/;
@cloud = qw/ясно малооблачно облачно пасмурно дождь ливень снег снег гроза N\/A -/;
@direct = qw/С СВ В ЮВ Ю ЮЗ З СЗ/;
for(;;) {
$weatherxml=get("http://informer.gismeteo.ru/xml/27612_1.xml") || die "xml-ошибка\n";
$weather=XMLin($weatherxml);
$strall = "$option\ПОГОДА - Москва\n\n";
@time = localtime();
foreach (@{$weather->{REPORT}{TOWN}{FORECAST}}) {
  if ($_->{day} == $time[3]) {$userdate = "Сегодня";} else {$userdate = "Завтра";}
 $str="$option$userdate [$tod[$_->{tod}]]\n$option\Температура: $value$_->{TEMPERATURE}{min}..$_->{TEMPERATURE}{max} $option\C\n$option\Осадки: $value$cloud[$_->{PHENOMENA}{cloudiness}], $cloud[$_->{PHENOMENA}{precipitation}]\n$option\Ветер: $value$direct[$_->{WIND}{direction}] [$_->{WIND}{min}-$_->{WIND}{max} м/c]\n\n";
  $strall .= $str;
}
$strall .= ($option . "gismeteo.ru\n");
print $strall;
sleep 60*60;
}
```

› **Прогноз погоды исчезнет ровно через пять секунд (или раньше, если вы перелистнете страницу).**

В скрипте используются два модуля, которые необходимо установить заранее: LWP::Simple и XML::Simple. Первый применяется для скачивания файла с web-сервера, а второй – для разбора полученного XML. Чтобы установить эти модули, выполните в терминале следующую команду:

```
# perl -MCPAN -e shell
```

Запустится оболочка, в которой нужно набрать следующее:

```
cpan> install LWP::Simple
cpan> install XML::Simple
```

Разумеется, это универсальный способ установки, и если в вашем дистрибутиве Linux некоторые модули можно добавить напрямую через менеджер пакетов, воспользуйтесь этой возможностью.

Теперь рассмотрим параметры *Dzen*, которые мы использовали в нашей программе:

» **-l** – включает ведомое окно, в качестве значения задается количество строк. У нас их 22.

» **-sa** – задает режим выравнивания текста в ведомом окне, значения аналогичны ключу **-ta**.

» **-tw** – указывает длину заголовка.

» **-e** – очень важный ключ, который сопоставляет события и действия. В данном примере использованы пары **entertitle=uncollapse** и **leavetitle=collapse**. При наведении ведомое окно разворачивается, при отведении – сворачивается. Полный список событий и действий, так же как и коды форматирования, можно прочесть только в readme-файле *Dzen*, в man представлены лишь основные ключи. Для просмотра описания наберите:

```
$ bzcat /usr/share/doc/dzen-[версия]/README.bz2 | less
```

В первом примере мы использовали *Dzen* как однострочную панель, в этом – как разворачивающийся список. Давайте немного изменим текущий скрипт. Уберём бесконечный цикл, допишем к параметрам *Dzen* ключ **-p 5** (задержка 5 секунд перед закрытием *Dzen*), добавим событие **-e 'onstart=uncollapse;'** (развернуть окно при запуске), уберем параметр длины заголовка, чтобы его ширина соответствовала ширине ведомого окна, и отредактируем координаты. Теперь запустим нашу программу. Появится окно с прогнозом погоды, которое закроется через 5 секунд. В данном случае *Dzen* будет работать в режиме всплывающих окон. Такой режим очень удобно использовать для задач, которые не требуют постоянной видимости – например, для вывода календаря; а скрипт можно вызывать из меню вашего оконного менеджера, по нажатию горячей клавиши или по любому другому событию.

Вам пришло SMS!

Мобильный телефон, который всегда должен быть рядом, очень часто оказывается в куртке в прихожей, где его совсем не слышно. Результат – несвоевременно прочитанные сообщения и пропущенные вызовы. В настоящее время редкий сотовый не имеет возможности соединения по Bluetooth. Если настроить Linux на автоматическую связь с телефоном через виртуальное COM-устройство (см., например, **LXF93** – http://bit.ly/9GD2ck), то можно будет посылать мобильному телефону AT-команды и получать результат, разобрав который, выводить требуемую информацию на панель *Dzen* на рабочем столе. Конкретизируем задачу: пусть нам нужна информация о состоянии счёта, статус телефона (если это «Звонок», есть шанс успеть добежать и ответить), а также счетчик пропущенных вызовов и текстовых сообщений. Список доступных AT-команд можно найти в Сети; многие из них универсальны и подходят для любой модели телефона, но лучше поискать конкретно для своей. Вот что получится в итоге (**mobile.pl**):

```
#!/usr/bin/perl
use Device::Modem;
$option="^fg(white)";$value="^fg(#4F95DE)";$critical="^fg(red)";
$wait = "$value\Ожидание"; $ring ="$critical\Звонок";
```

```
$status = $wait;$calls = 0;$sms = 0; $bc = $sc = $cc = $value;
open (DZEN, "| dzen2 -fn -*-fixed-*-r-*-*-12-*-*-*-*-*-*-* -bg
'#303030' -x 220 -y 0 -w 420 -ta r") || die "dzen2-ошибка!\n";
select (DZEN); $| = 1;

$modem = new Device::Modem(
port => "/dev/rfcomm0",
log => 'file,/dev/null',);
$modem->connect(baudrate=>9600);
foreach (qw/ATZ AT+CNMI=2,3 AT*CPI=2/) {
$modem->atsend($_ . Device::Modem::CR);
$modem->answer($Device::Modem::STD_RESPONSE,1000);
}
$modem->atsend('AT+CUSD=1,"#102#"' . Device::Modem::CR);
$answer = $modem->answer
($Device::Modem::STD_RESPONSE,10000);
if ($answer =~ /CUSD:\ 0.*"\D+?([0-9.-]+)/) {
$bc = $value;
$money = $1;
if ($money <= 0.5) {$bc = $critical;}
}

for (;;) {
$answer = $modem->answer(qr/CPI|CMTI/,10000);
if (defined($answer)) {
   if ($answer =~ /CPI: 1,2/) {$status = $ring;$calls += 1; $cc =
$critical;}
     if ($answer =~ /CPI: 1,1/) {$status = $wait;}
     if ($answer =~ /CMTI/) {$sms +=1;$sc = $critical;}
}
print "$option\[ТЕЛЕФОН] Баланс: $bc$money$option | Статус:
$status$option | Звонков: $cc$calls$option | SMS: $sc$sms \n";
}
```

Для связи с COM-устройством мы используем модуль Device::Modem – установите его аналогично двум предыдущим.

Пара пояснений по использованным здесь AT-командам:

» AT+CUSD=1,"#102# – послать USSD-запрос оператору, аналогично набору #102# на мобильном телефоне (у моего оператора – информация о текущем счёте); но в данном случае ответ выводится в терминал и с помощью регулярных выражений из него выбирается информация о балансе счёта.

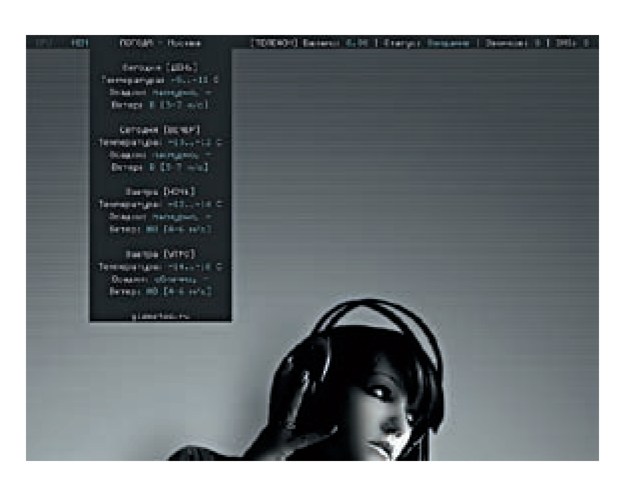
» AT+CNMI=2,3 – разрешить индикацию прихода SMS; по умолчанию выключено.

» AT*CPI=2 – аналогично предыдущей функции, но только для звонков.

Таким образом, после инициализации и получения сведений о балансе программа переходит в бесконечный цикл и опрашивает мобильный телефон, выявляя новые события. Разумеется, в таком виде она не совсем пригодна для постоянного использования: необходимо проверять активность модема, обрабатывать сообщения об ошибках и т. д. В любом случае можно посмотреть: наверняка кто-нибудь уже делал нечто подобное. Например, есть консольная программа *gammu*, которая может избавить вас от необходимости искать описание AT-команд в Интернете.



› *Dzen*-апплет для телефона: 1 и 2 — в режиме ожидания и при входящем сообщении, 3 — альтернативный компактный вид.



› Панель выглядит как единое целое: сразу и не скажешь, что это результат работы четырех независимых скриптов.

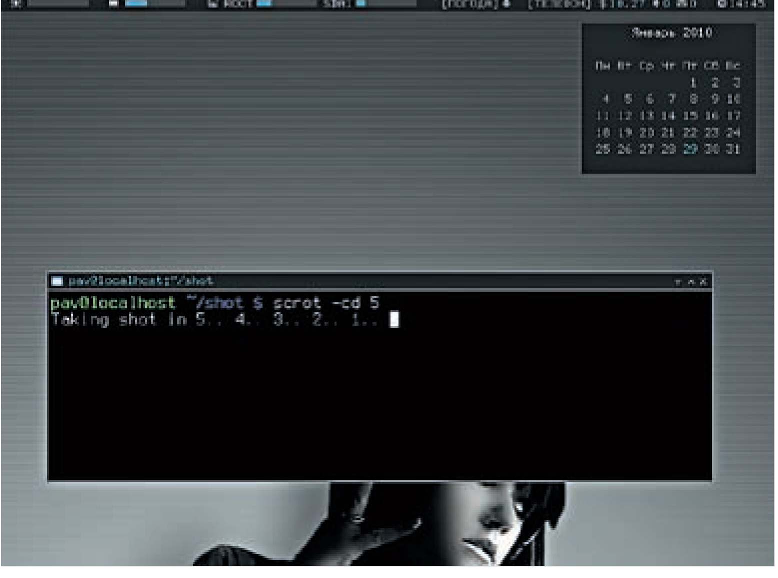
Ура, готово!

Итак, посмотрим, что же у нас получилось. Скомпонованные вместе, все описанные выше скрипты должны сформировать единую информационную панель. Сделаем каждый сценарий исполняемым и запустим:

```
$ chmod a+x cpu.pl mem.pl meteo.pl mobile.pl
$ ./cpu.pl & ./mem.pl & ./meteo.pl & mobile.pl &
```

Если результат вас удовлетворит, можно прописать их запуск в автозагрузку вашей сессии. Но учтите: окна *Dzen* располагаются поверх всех остальных, их нельзя перемещать и менять размер. Так что если вы решили использовать *Dzen* в качестве панели, которая постоянно на виду, учтите эти нюансы. В своем оконном менеджере *Openbox* (**LXF124**) я просто установил в настройках рабочего стола величину верхнего поля, достаточную для размещения панели. Остальная работа – это уже творчество, ограниченное скромными (пока) возможностями *Dzen*. Но, как уже говорилось, проект не стоит на месте и постоянно развивается. Последние новости, документацию и другие полезные ресурсы вы всегда найдете на странице программы, доступной по адресу http://sites.google.com/site/gotmor/dzen. **LXF**

› Изображенные здесь дополнительные апплеты (индикатор съемных устройств, часы и календарь) вы теперь сможете написать самостоятельно.



Популярно о моделировании

Пусть вы и не инженер – свободное ПО позволяет заглянуть в тайны этой профессии всем желающим. В качестве примера **Леонид Алифанов** построит модель и выполнит расчет плотины Саяно-Шушенской ГЭС.



Наш эксперт

Леонид Алифанов
Ведущий инженер ЗАО «СибПСК», г. Красноярск (www.sibpsk.ru) и давний любитель свободного ПО.

В семействе открытых пакетов программ для моделирования и расчетов методом конечных элементов (МКЭ) немало славных представителей. Есть такие промышленные тяжеловесы, как Code_Aster® от Электрисите де Франс, есть и небольшие удачные программы, «заточенные» авторами для специфических инженерных и научных задач.

К сожалению, многие проекты такого рода со временем затухали, оставаясь без поддержки или интереса со стороны пользователей. Но практически безотказным условием их успешности является авторство в них практикующих инженеров, создающих расчетные инструменты для собственных нужд. Один из таких пакетов – *CalculiX* (www.calculix.de) разработан двумя немецкими конструкторами авиадвигателей – Гвидо Дхондтом [Guido Dhondt] и Клаусом Виттигом [Klaus Wittig].

Пакет создавался для прочностных и тепловых расчетов деталей и узлов авиадвигателей, поэтому основной упор в нем сделан на объемные конечные элементы, причем наиболее точные – с промежуточными узлами.

Зачем все это?

Может ли вообще понадобиться простому смертному рассчитывать лопатки турбин в авиадвигателе? Ну, даже если он и захочет это сделать, будет лучше поучиться лет 5 и набираться опыта лет 10. Но бывают случаи, когда хочется «пощупать» конструкцию самому. В конце концов, если любая профессия – в действительности заговор против непосвященных, то иногда все же хочется представлять себе контуры этого «заговора» в общих чертах.

Кто не обсуждал в последнее время надежность плотины Саяно-Шушенской ГЭС? Можно ли было предположить, что книга Валентина Брызгалова – скучное, на первый взгляд, производственное издание «Из опыта создания и освоения Красноярской и Саяно-Шушенской гидроэлектростанций» – станет цитируемой повсеместно?

Это гидротехническое сооружение поражает воображение уже своим общим описанием: «напорный фронт образует уникальная бетонная арочно-гравитационная плотина высотой 245 м, длиной по гребню 1066 м, шириной по основанию 105,7 м, а по гребню – 25 м …». 245 метров – как две красноярские телебашни с антеннами, поставленные одна на другую!

Каждый решал в школе элементарные задачки по физике, в которых силу делят на площадь, получают давление и сравнивают его с допустимым. Школьный курс дает самое приблизительное представление о прочностных расчетах, увязывая вопрос о прочности конструкции с действующими внутри нее напряжениями (давлением внутренних сил в материале). Задача и состоит в определении их в общем случае.

Что же такое метод конечных элементов, реализованный в *CalculiX* и в тысячах подобных программ? Объемная конечно-элементная модель представляет собой некое тело, заполненное тетраэдрами или призмами, образующими сетку. Каждый элемент имеет узлы, установленные как минимум в угловых точках, причем в сплошном массиве для смежных элементов эти узлы общие. Для тела определяются граничные условия – прикладываются известные силы, давления и закрепления. Для каждого «элементарного кусочка» составляются уравнения, учитывающие его равновесие и зависимости между неизвестными деформациями и напряжениями. В результате объединения систем уравнений и расчета, эти неизвестные напряжения на элементарных площадках внутри тела могут быть определены.

Приступим к работе

Итак, восстановив в памяти самые элементарные сведения из школьной физики, в данной статье мы с вами попробуем подойти к плотине не с голыми руками, но вооружившись *CalculiX*. Если даже вы и не освоите программу, то, по крайней мере, сможете понять общую логику рассуждений и посмотреть на ре-

Немного о CalculiX

Кроме объемных, в *CalculiX* имеются также элементы для плоских задач (плоско-напряженное и плоско-деформированное состояние), стержневые и оболочечные. Касаемо двух последних типов стоит заметить, что реализованы они немного нетривиально – на основе все тех же «объемников». Поэтому результаты расчета над стержнями чувствительны к соотношению габаритов отдельных КЭ, и для получения правильного результата нужно использовать не слишком вытянутые элементы. Оболочечные, переделанные из объемных, выдают результаты по теории толстых плит и считают только напряжения (внутренние усилия, по которым подбирается армирование железобетонных конструкций, не выдаются).

С *CalculiX* связан единственный в своем роде интернет-проект www.onlinefeasolver.com – с его помощью желающие могут не только рассчитать конструкцию или деталь, но и сделать т.н. топологическую оптимизацию изделия. Загрузив бесформенный кусок материала, вы получите конструкцию оптимальной формы: специальный скрипт анализирует силовые линии и, обеспечив сопротивление прилагаемым нагрузкам, удаляет избыток материала.

зультаты. Пока вы компилируете исходные тексты или скачиваете RPM-пакет с SourceForge.net, я запущу специальную сборку Ubuntu с предустановленными программами для моделирования с www.caelinux.com.

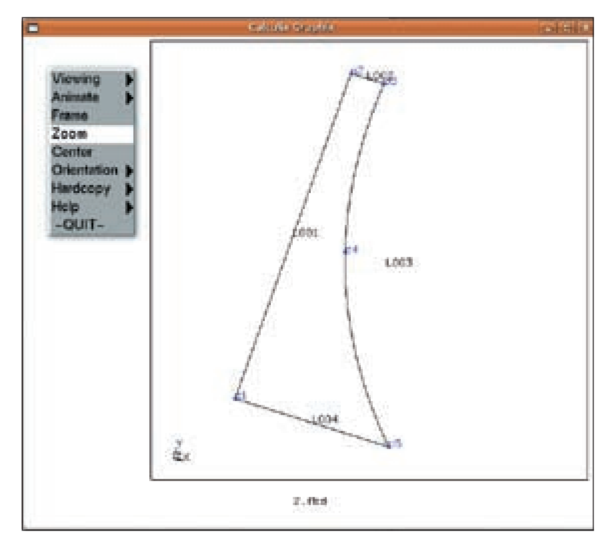
CalculiX – пакет, состоящий из двух модулей: препостпроцессора CGX и решателя CCX. Приступим к построению так называемой твердотельной модели, внутри которой будут заключены конечные элементы. Этот процесс осуществляется в CalculiX CGX (CalculiX Graphics). Для запуска программы в режиме построения, наберем в консоли

```
cgx -b <имя_файла>.fbd
```

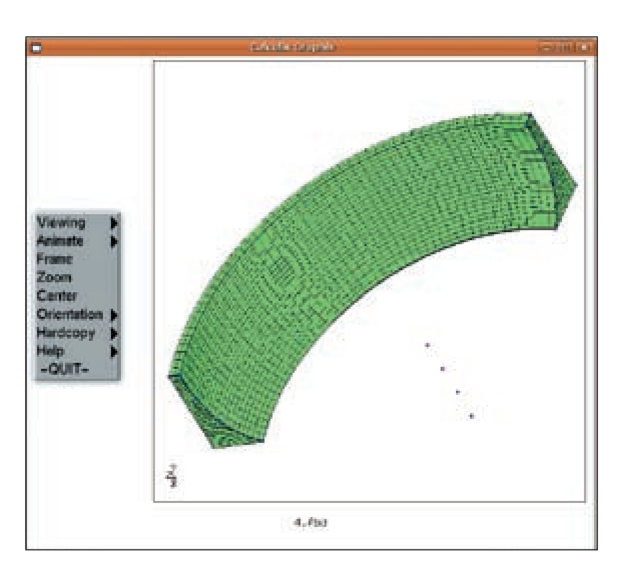
Возникнет графическое окно CGX (рис. 1). Нам нужно ввести точки, затем соединить их линиями, сгенерировать внутри контура поверхность и вытянуть ее вдоль дуги в массивную арку. Процесс построения может осуществляться интерактивно (вводом команд при открытом графическом окне и последующей записью в FBD-файл в рабочем каталоге *CalculiX*) или в текстовом редакторе. Детально описывать действия и команды я не буду, тем более что руководства для начинающих, в том числе и на русском языке, доступны в Сети. А вот на коде, который описывает модель, остановлюсь несколько подробнее.

```
pnt p1    -10     0       0
pnt p2    -10     245     0
pnt p3    15      245     0
pnt p4    19.1    163.3   0
pnt p5    39.3    81.7    0
pnt p6    105.7   0       0
pnt p7    600     0       0
pnt p8    600     1       0
seqa seq PNT p3 p4 p5 p6
LINE L001 p1 p2 48
LINE L002 p2 p3 4
LINE L003 p3 p6 seq 48
LINE L004 p6 p1 12
gsur S001 + BLEND - L004 - L003 - L002 - L001
seta sec s S001
swep sec dam rot p7 p8 102 100
elty all he20
mesh all
```

Команды **pnt** – это ввод точек в декартовой системе координат (размеры в метрах взяты приблизительные). **LINE** – линии (три отрезка и один сплайн по точкам, определенным командой seqa). В конце каждой команды стоит число, задающее количество конечных элементов вдоль геометрического примитива «линия».



› Рис. 1. Отображение точек и линий, ограничивающих фигуру — сечение плотины.



› Рис. 2. Конечно-элементная модель плотины после вытяжки.

Команда **gsur** отвечает за построение поверхности, а **swep** – это вытяжка поворотом вокруг вектора p7–p8 на угол 102°. Последние две строчки: **elty** – задание типа элемента и **mesh** – команда на генерацию сетки.

То, что получилось в результате, вы можете увидеть на рис. 2.

Для отображения заданных примитивов в CGX используются команды **plot** и **plus**, вводимые при активном графическом окне. Например **plot e all** означает «отобразить все элементы», а **plus na all** – «добавить к изображению узлы с их номерами». В общем, если вы все это проделаете, то процесс построения в общих чертах будет вам понятен.

Сразу же представляю себе реплику неискушенного читателя: «В XXI веке строить модели командами – да еще в заумном текстовом файле!» Возражения на этот счет вот какие: во-первых, команды выбраны таким образом, чтобы удобно было генерировать их автоматически из скриптов для построения параметрических моделей. Ведь инженер заранее не знает точные параметры своей конструкции или детали, а перестраивать модель, каждый раз тыкая указателем мыши в лубочные формы на экране, утомительно. Во-вторых, особенности построения модели предопределяют конечно-элементную сетку, которую генерирует CGX – она качественней, в смысле точности, автоматических треугольников и тетраэдров, создаваемых другими аналогичными программами, и этот факт в некоторых задачах может иметь важное значение.

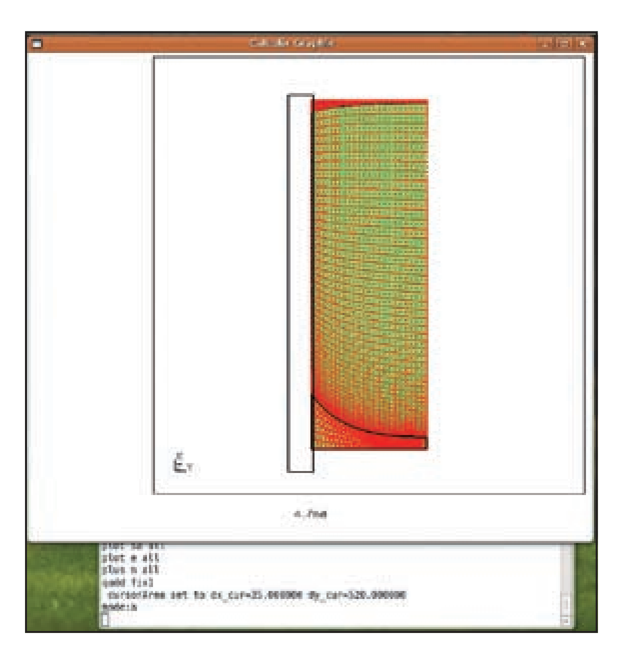
На экспорт

Итак, сетка построена, координаты узлов и элементов известны и могут быть записаны в отдельный файл; но работа в CGX пока не завершена. Модуль CalculiX CCX (решатель) работает с сетками в специфическом текстовом формате. Узлы задаются координатами по номерам (сквозная нумерация не обязательна); в заголовке указывается имя группы узлов, которой пользователь оперирует в дальнейшем. Именованные группы узлов и элементов необходимы для задания нагрузок и закреплений, и они также формируются в CGX и записываются в текстовые файлы командой **send** с заданными ключами.

Созданию именованных групп служат команды **qadd** и **qrem**. Перед созданием группы из определенных примитивов следует отобразить их через plot/plus. Для примера продемонстрирую создание группы fix1 узлов подошвы плотины.

```
plot n all
qadd fix1
```

После ввода последней команды на кончике указателя мыши возникает маленький квадратик, которым в графическом окне можно захватывать и выделять нужные узлы, точки и т. д. Чтобы увеличить его размеры, нужно дважды последовательно »



› Рис. 3. Выделение узлов модели с помощью рамки.

установить курсор в диагонально противоположные точки воображаемого прямоугольника выделения и дважды ввести **r** с клавиатуры (рис. 3). Далее, если вы собираетесь выделять сразу много примитивов, попадающих внутрь рамки, введите с клавиатуры **a**, после чего отметьте необходимое, перемещая прямоугольник по экрану и вводя ключевой символ требуемого вам примитива (для узлов – **n**, для элементов – **e**; для поверхностей элементов – **f**, и т. д.). Команда завершается вводом **q**. Если в группу при выделении попали лишние примитивы, они могут быть удалены аналогично, при помощи команды **qrem**.

> «Вы можете это проверить, построив модель самостоятельно.»

Отобразить созданную группу можно, например, так:

```
plus n fix1 g
```

где **g** – это green (зеленый). В CGX использованы следующие цветовые ключи: **r** – red (красный), **g** – зеленый, **k** – black (черный), **b** – blue (голубой), **m** – magenta (пурпурный).

Отправить созданную группу (зафиксированные номера узлов, которые будут использованы в дальнейшем) в текстовый файл можно командой

```
send fix1 abq nam
```

В плоском случае

Возвратимся к нашему примеру: давайте сначала просчитаем плоскую задачу. Мысленно вырежем из плотины метровый сегмент и зададим основную для него гидростатическую нагрузку со стороны верхнего бьефа и закрепления по подошве. Гидростатическое давление, которое увеличивается с глубиной, вычисляется как произведение удельного веса воды на глубину погружения рассматриваемой площадки.

Наберем в текстовом редакторе файл следующего содержания:

```
*INCLUDE, INPUT=all.msh
*INCLUDE, INPUT=fix1.nam
*MATERIAL,NAME=EL
*ELASTIC
0.25E+05, 0.200000
*DENSITY
0.024
*SOLID SECTION,ELSET=Eall,MATERIAL=EL,
*BOUNDARY
Nfix1,1,2
*STEP
*STATIC
*DLOAD
Eall,GRAV,1.,0.,-1.0,0.
*DLOAD
1,P4,2.3479
…
24,P4,0.1021

*NODE FILE
U
*EL FILE
S
*END STEP
```

Команда *INCLUDE здесь подключает к расчету файл сетки **all.msh** и файл **fix1.nam** с группой узлов Nfix1 для задания закреплений по подошве. *ELASTIC – задает упругий материал с модулем Юнга E=$2{,}5\cdot10^4$ МПа и коэффициентом Пуассона μ=0,2; *DENSITY – объемный вес бетона (в МН/м^3); *SOLID SECTION – это присвоение заданных параметров группе элементов Eall (в которую входят все элементы); *STEP ... *END STEP – начало и конец описания рассматриваемого шага нагрузки (варианта нагружения, в данном случае – единственного). *BOUNDARY – здесь задаются закрепления либо вынужденные перемещения групп узлов; *DLOAD – активируется учет объемных сил от собственного веса элементов, и этой же командой прикладывается гидростатическое давление (например, 1,P4,2.3479 – давление в МПа на поверхность P4 элемента номер 1 и так далее). *NODE FILE, *EL FILE – команда на выдачу определенных результатов (узловых перемещений и напряжений в элементах).

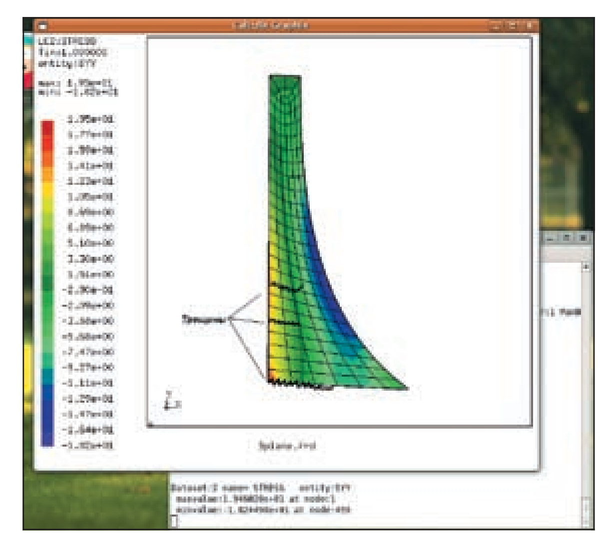
Дадим в консоли команду

```
ccx plane
```

и затем, когда расчет будет окончен, запустим модуль CGX в режиме постпроцессора:

```
cgx -v plane.frd
```

Отображаем с помощью контекстного меню напряжения SYY, в МПа, действующие по площадкам, параллельным основанию, и получаем картинку, представленную на рис. 4. Положительные напряжения соответствуют растяжению, отрицательные – сжатию. Растягивающие напряжения получаются слева, по напор-



› Рис. 4. Отображение напряжений в CGX.

ной грани. Бетон на растяжение работает плохо, и в этой зоне после наполнения водохранилища возникли трещины, из-за того, что при проектировании расчеты выполнялись приближенными методами и растягивающие напряжения были недооценены проектировщиками. Это та самая проектная ошибка, о которой много писали и говорили – те трещины, которые впоследствии заделывали специальными смолами приглашенные французские специалисты.

Решенная нами плоская задача рассматривает вариант, при котором плотина работает только как гравитационная. Если бы по какой-то причине плотина не смогла передавать горизонтальные усилия на берега, то трещины прорезали бы массив бетона на всю толщину, что привело бы по меньшей мере к ее сдвигу под нагрузкой, так как при наличии трещин проектные предположения о величине коэффициента трения и о сцеплении с основанием не работают. В то же время вода, просачиваясь в трещины, создавала бы дополнительное давление, направленное вверх, против веса дамбы. В связи с этим, излишне говорить, что вес намерзшего этой зимой льда, которым нас пугали в последнее время, на самом деле не только не приносит вреда, но даже полезен, поскольку увеличивает пригруз и суммарный момент сил, удерживающих плотину от опрокидывания. Вы можете это проверить, построив модель и приложив вертикальную нагрузку самостоятельно. Само собой, даже напрашивается необычное рацпредложение – наморозить льда как можно больше и сохранять его летом, теплоизолировав снаружи и пропустив внутри по трубкам фреон (ну или что-то в этом роде).

А вот высказывание о том, что плотина работает «как распорка» (точнее, как арка, то есть держится за счет передачи распора на берега) действительности полностью соответствует, в связи с чем во многих учебниках «гравитационная» в названии сооружения уже и не упоминают.

Также становится понятным заявление специалистов ГЭС о том, что другая известная проблема – с водобойным колодцем, не выдерживающем паводка – не столь актуальна, как это себе представляют (колодец – отдельное сооружение, а небольшой подмыв основания не особенно угрожает арочной работе).

Влияние берегов

Теперь посчитаем пространственную модель, то есть попробуем учесть работу плотины как арки. За неимением точных данных, я просто удалил «на глазок» элементы по берегам, жестко заделав плотину по бокам и в основании. Расчетный INP-файл следует переделать. Поверхностная гидростатическая нагрузка была задана мной на заранее отмеченные и сохраненные поверхности объемных конечных элементов типа C3D20:

```
*DLOAD
*INCLUDE,INPUT=f0.dlo
...
*INCLUDE,INPUT=f12.dlo
```

Здесь **f0.dlo** и так далее – файлы со значениями нагрузок, записанные из CGX и сохраненные в рабочем каталоге.

Производим расчет и отображаем напряжения SYY. Как мы видим (рис. 5), растягивающие напряжения по напорной грани остаются.

Порядок величин перемещений и «арочных» сжимающих напряжений (в *CalculiX* они просматриваются как главные напряжения SP3, наименьшие с учетом знака или наибольшие сжимающие со знаком «минус») в моем расчете приблизительно совпал с опытными значениями, полученными в результате измерений. Однако, поскольку сведений, приведенных в открытых источниках, недостаточно для построения точной модели, я не буду останавливаться на анализе этих величин.

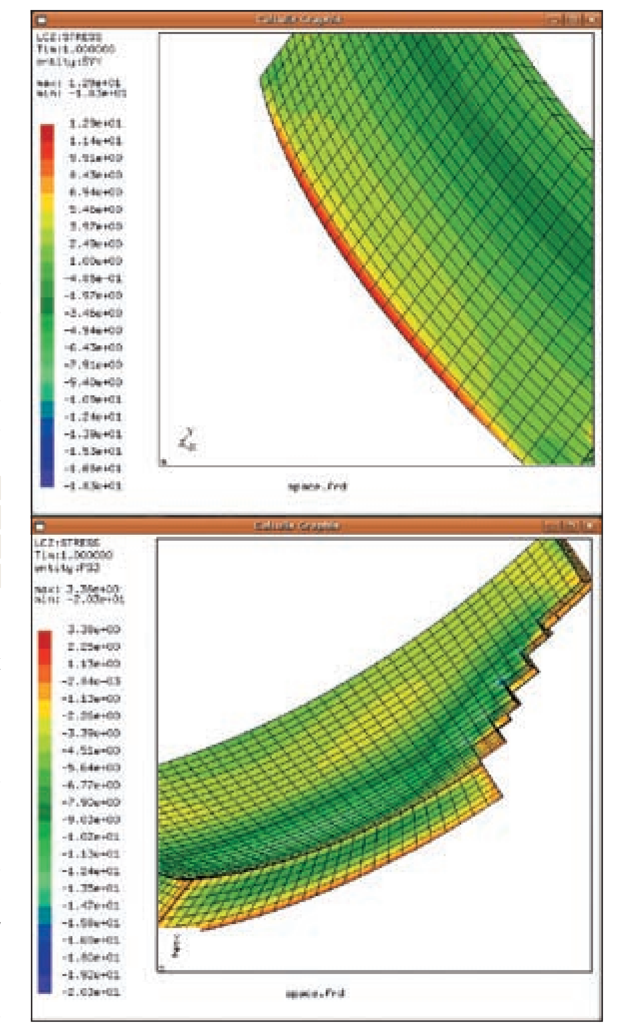
Во время проектирования плотины, численные методы, такие как МКЭ, уже внедрялись во многих странах, однако инженеры-гидротехники их проигнорировали. Статический расчет производился так называемым «неполным методом пробных нагрузок (арок-консолей)». За счет арочного эффекта планировалось снять всего 25 % от расчетного гидростатического давления.

Вообще, стоит отметить, что приведенный расчет, выполненный в линейной постановке без учета трещин, совместной работы с основанием и т. д., дает очень приблизительные, оценочные результаты. Они не помогут ответить на вопрос, насколько надежно или нет бетонное сооружение. О существовании приближенной к действительности математической модели плотины СШГЭС можно прочесть в уже упомянутой книге Валентина Брызгалова.

Назначение данной статьи состоит лишь в популяризации открытого программного обеспечения, но особенно важно то, что использование таких пакетов, как *CalculiX*, позволяет оценивать техническую информацию как положено – с точки зрения элементарных научных знаний, и не забивать голову фантастическими измышлениями, гуляющими в прессе. LXF

Сделаем уточнение

На самом деле, выдаваемые *CalculiX* максимальные значения растягивающих напряжений не верны. И дело не в программе. В точке, где они достигаются (рис. 4), исследуемые напряжения бесконечно велики, как значение функции $y = 1/\sqrt{x}$ при $x \to 0$. Это так называемая «особая точка». Такие же точки имеются в вершинах трещин, поэтому для исследования их стабильности используются другие величины, которыми оперирует сравнительно молодая наука – механика разрушения. Факт тот, что в данном случае серьезные растягивающие напряжения имеются на значительном объеме массива плотины, и кроме трещины на контакте с основанием есть и другие – выше него, схематически показанные на рис. 4.



› **Рис. 5. Отображение результатов в CalculiX CGX.**

Go: на языке классиков

Языков программирования и так развелось немало – стоит ли изобретать еще один? **Андрей Боровский** разбирается, смогли ли в Google найти ответ на этот извечный вопрос.



Наш эксперт

Андрей Боровский
Постоянно открывает для себя что-то новое, изучая языки, библиотеки и прочие инструменты программиста по заданию **LXF**.

Не знаю, как вы, а я по-прежнему придерживаюсь мысли, что развитие языков программирования происходит под влиянием теоретических идей, а не аппаратных новинок. Абстрактные типы данных, ООП, безопасные указатели, сборка мусора – все это было придумано довольно давно и независимо от прорывов в области оборудования (которое иногда не поспевало за полетом мысли теоретиков). Как это часто бывает в научной среде, стагнация теории заставляет исследователей искать вдохновения в смежных областях. Применительно к языкам программирования, многие обращаются к параллелизму, который теперь стал доступен пользователям домашних компьютеров. По моему глубокому убеждению, многопоточность совершенно ортогональна структуре языка программирования. Многопоточные приложения можно одинаково успешно реализовывать на любом языке, в котором есть концепция подпрограммы, о чем и свидетельствуют многопоточные программы, написанные на C, C++, Java, C#, Pascal и Visual Basic. Тем не менее, разработчики новых языков часто говорят о том, что их детище адаптировано для параллелизма особенно хорошо. Не является исключением и Go. Впрочем, стоит отметить, что для Go подобное утверждение в общем оправдано. Разработчики Go не внесли нового в теорию параллельного программирования, но создали инструмент, который, будучи потенциально так же эффективен, как C, существенно упрощает процесс написания многопоточных программ.

Тень Google

Вряд ли очередной язык заслужил бы столько внимания, если бы не поддержка со стороны Google и лично Роба Пайка [Rob Pike], легендарного участника разработки ОС Unix и Plan 9; от Plan 9 Go унаследовал некоторые особенности. На сайте проекта (http://golang.org) утверждается, что Go планировалось использовать для написания серверных приложений Google (а они, вероятно, принадлежат к самым быстродействующим и стрессоустойчивым приложениям в мире). Если бы Go преуспел в таком качестве, пожалуй, уже этого было бы достаточно, чтобы все забросили C++ и Java. Но, согласно заявлениям разработчиков, «Go еще не созрел для использования в серверных приложениях в широком масштабе, но мы (разработчики) трудимся над этим». Оценить, насколько хорош Go для написания серверов, можно уже сейчас. Web-сервер, обслуживающий сайт golang.org – это написанная на Go программа *godoc*, которая доступна, как и весь пакет Go, любому программисту, работающему под *nix. Последний факт, кстати, заслуживает особого внимания. В настоящее время Go существует для Linux, FreeBSD и Darwin; на создание порта для Windows, по словам разработчиков, не хватает ресурсов. Однако против ОС от Microsoft они, в принципе, ничего не имеют и с радостью примут помощь в переносе Go на самую популярную платформу.

› **Логотип Go чем-то неуловимо напоминает Гленду из Plan 9, и неспроста.**

Возвращаясь к программе godoc, отметим, что ее основное назначение – генерировать документацию для программ, написанных на Go. Однако, будучи запущена с ключом **--http**, программа превращается в web-сервер и начинает выдавать не простой текст, а web-страницы (как вам это нравится?).

Найдем пять отличий

Чем же Go отличается от привычных нам языков, таких как C++? Прежде всего, здесь отсутствует понятие класса. Различные проблемы, которые в C++ решаются множественным наследованием, объявлением методов-друзей, шаблонами и другими способами, в Go решаются с помощью интерфейсов.

Массивы в Go всегда передаются по значению (прямо Паскаль какой-то). При этом некоторые другие типы данных (например, хэш-таблицы, которые тоже являются встроенными типами языка) передаются по ссылке. Концепция указателей существует, но их арифметика не реализована (что естественно для языка, автоматически управляющего памятью). Для экономии циклов процессора и ОЗУ при работе с функциями можно использовать фрагменты массивов (сечения, slices). Между прочим, в стандартизированном Паскале такая возможность тоже есть. Строки также являются встроенным типом, причем неизменяемым (как в Java/C#). Потоки и каналы (средства обмена данными между потоками) также реализованы как встроенные конструкции. Все преобразования типов в Go выполняются явным образом.

Инструментарий

Устанавливать Go лучше всего непосредственно из интернет-репозитория (инструкции вы найдете по адресу http://golang.org/doc/install.html). Для сборки Go вам понадобятся такие утилиты, как *bison* и *gawk*. У меня этот процесс закончился обнадеживающим сообщением «0 known bugs; 0 unexpected bugs». Вы, наверное, ждете, что после сборки Go у вас в системе появится команда **go**, которая и будет вызвать компилятор? Ничего подобного. Прежде всего, разработчики Go считают, что компилятор еще не созрел до уровня общесистемной команды Linux, так что по умолчанию все двоичные файлы установятся в **~/bin** (этот каталог необходимо создать либо до, либо в процессе установки Go). Скомандовав **ls ~/bin**, вы увидите много файлов, но среди них опять-таки не будет **go**. Содержимое этой директории будет зависеть от архитектуры вашей системы. На моей amd64 там расположены:

» *6g* – компилятор Go для 64-битных платформ Intel/AMD (32-битный вариант именуется 8g). На вход компилятора подается файл исходных текстов (с расширением **go**), а на выходе мы получаем объектный код с расширением из одной цифры (той, с которой начинается имя компилятора).

» *6l* (и, соответственно, 8l) – компоновщик Go, превращающий объектный код в исполняемый файл (по умолчанию, **6.out** или **8.out**). Такая необычная система имен берет начало в традициях операционной системы Plan 9 (**LXF126/127**).
» *6c* (*8c*) – компилятор C, изначально разработанный для Plan 9 (зачем он нужен – непонятно, так как для сборки Go использует *GCC*). Интересующиеся могут найти документацию по нему здесь: http://plan9.bell-labs.com/magic/man2html/1/2c.
» *6a* (*8a*) – ассемблер родом из той же ОС.
» *ebnflint* – это не русское ругательство в адрес английского пирата, а инструмент для работы с грамматиками в формате EBNF.
» *godefs* – инструмент для тех, кто портирует среду времени выполнения Go на новые платформы.
» *gofmt* – программа, форматирующая исходные тексты на Go (расставляет отступы и прочее).

Помимо этого, наличествуют *6nm* (аналог *nm*), *6prof* (*gprof*), *6cov* (*gcov*), уже упомянутый *godoc* и *cgo* – команда, предназначенная для создания пакетов Go, вызывающих функции, написанные на C. Наконец, утилита с приятным украинскому уху названием *gopack* – это всего лишь аналог *ar* для Go.

Все команды Go высокомерно игнорируют ключи типа **--h** и **--help**; при этом man-страницы для них тоже отсутствуют. Получить краткое описание работы той или иной утилиты можно либо с помощью команды

```
godoc <имя_утилиты>
```

либо по адресу http://golang.org/cmd/.

Let's Go!

Ознакомившись с инструментарием, перейдем к знакомству с самим языком программирования. Now (не могу удержаться от англоязычного каламбура) let's go say Hello World!

Наша первая программа (**hello.go**) выглядит так:

```
package main
import "fmt"
func main() {
fmt.Printf(" Здравствуй, Мир!\n Hello, world!\n")
}
```

Для компиляции и сборки программы командуем, соответственно,

```
./6g hello.go
./6l hello.6
```

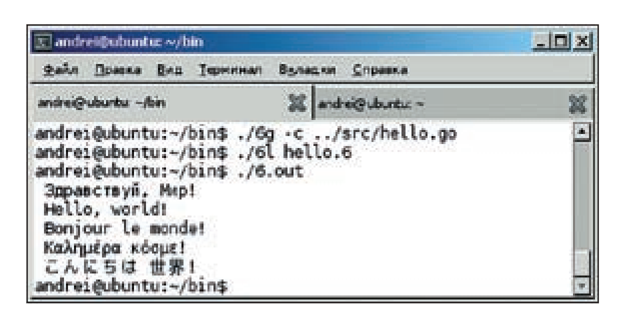
а для выполнения –

```
./6.out
```

Результат многоязычного приветствия можно увидеть на экранном снимке.

Не стоит относиться к программам «Здравствуй, Мир!» свысока. Внимательный взгляд на них может многое рассказать о языке. Программа начинается с ключевого слова package, которое объявляет пакет main. Package (пакет) является минимальной единицей компиляции в Go. Он также образует отдельное пространство имен. Попросту говоря, вы складываете в пакет определения функций, структур и объявления переменных – возможно, для того, чтобы использовать их потом в других пакетах. Пакеты могут состоять из одного или нескольких исходных файлов. Больше всего пакеты Go похожи на модули Паскаля (концепция заголовочных файлов в Go отсутствует) с той разницей, что файл программы, как мы видели, тоже является пакетом.

Ключевое слово import указывает, что мы хотим импортировать в нашу программу пакет – стандартный fmt, который, помимо прочего, содержит функцию Printf(). Последняя работает как в C, но использует другие спецификаторы формата вывода переменных. Советую ознакомиться с ее описанием (**./godoc fmt Printf**). Также обратите внимание на то, что идентификаторы элементов, экспортируемых из внешних модулей, всегда начинаются с заглавной буквы.



› **Программа «Здравствуй, Мир!» в исполнении Google Go.**

Далее мы определяем функцию main(). В отличие от C и C++, в языке Go для объявления функций используется специальное ключевое слово func, как в Паскале. Это далеко не единственное сходство между Go и Паскалем, и дело тут не в языковых пристрастиях авторов. Паскалеподобные элементы Go преследуют важную (с точки зрения разработчиков) цель – повысить скорость компиляции. Лаконичная грамматика C заставляет компилятор выполнять много дополнительной работы по выяснению смысла языковых конструкций. Вспомогательные слова Паскаля и Go упрощают работу компилятора (хотя и вынуждают программиста напечатать несколько дополнительных символов). Как и в C, блоки операторов выделяются в Go фигурными скобками, но **;** как разделитель операторов не используется. Точнее говоря, грамматика языка ее допускает, но она не является обязательной. Если говорить еще точнее, синтаксический анализатор Go вставляет символ ; в конце каждой строки, которая завершает оператор. Это правило имеет интересные следствия. Если вместо

```
func main() {
```

мы напишем

```
func main()
{
```

то синтаксический анализатор преобразует этот фрагмент в

```
func main();
{
```

что, естественно, приведет к ошибке. Иначе говоря, открывающая скобка всегда должна находится на той же строке, что и заголовок функции. Это же правило относится и к другим операторам, использующим скобки. Например, если вместо

```
if a > b {
Printf("a > b")
}
```

написать

```
if a > b
{
Printf("a > b")
}
```

то функция Printf() будет вызываться всегда, независимо от значений a и b. При этом компилятор сообщит об ошибке, так как выражение

```
if a > b;
```

»

Бесконечный цикл за 5 секунд!

Вы, конечно, подумали о том, что может произойти, если вместо

```
for i < 10 {
```

мы напишем

```
for i < 10
{
```

В отличие от оператора

```
if i<10;
```

оператор

```
for i<10;
```

породит синтаксическую ошибку (а не бесконечный цикл).

Правда, компилятор укажет на ошибку не совсем там, где мы ожидаем, но мы, по крайней мере, будем знать, что в программе что-то не так.

Для создания бесконечного цикла, то есть такого, выход из которого осуществляется с помощью break или os.Exit() (например, цикла обработки сообщений оконной системы) можно использовать конструкцию

```
for {
...
}
```

является вполне законным (хотя и бесполезным). Будет выдано только предупреждение. Все это должно радовать последователей Linux-стиля оформления кода на C, однако стоит заметить, что в Go это правило не всегда очевидно и может стать источником трудновыявимых ошибок. Второе следствие заключается в том, что блоки операторов в Go нужно всегда заключать в фигурные скобки, даже если оператор всего один, как в примере выше.

Обратите внимание на то, что перед именем функции Printf() мы указываем имя пакета. Если интуиция программиста подсказывает вам, что существует способ избежать утомительных префиксов, то вы правы. При импортировании пакетов мы можем указать псевдоним, который затем будет использоваться вместо его имени. Например, если бы мы написали

```
import f "fmt"
```

то вызов функции Printf() выглядел бы так:

```
f.Printf(...)
```

Если же мы напишем

```
import . "fmt"
```

функции из пакета fmt можно будет вызывать вообще без префикса.

Еще одна «примочка» Go – использование в идентификаторах символов Unicode. Например:

```
func main() {
var Путь, Время float = 250, 100
var СредняяСкорость float = Путь/Время
Printf("Средняя скорость %f\n", СредняяСкорость)
}
```

У этого подхода к именованию есть свои сторонники и противники. Я не стану вмешиваться в их спор – замечу только, что с использованием Unicode все не так просто, как кажется на первый взгляд. Во многих алфавитах существуют составные символы, использование которых в Go запрещено. Также, согласно правилам Go, экспортируемые модулем идентификаторы должны начинаться с заглавной буквы, а таковые присутствуют не в каждом алфавите.

Обратите внимание на то, как в Go объявляются переменные. Здесь мы тоже наблюдаем сходство с Паскалем. Объявление начинается с ключевого слова var, а тип переменных следует за перечнем их имен. Если переменная инициализируется в момент объявления, тип можно не указывать. Например, встретив объявление переменной

```
var i = 10
```

компилятор сам присвоит переменной i тип int.

Сходство с C заключается в том, что мы можем инициализировать переменные в процессе их объявления. Помимо полной формы, которую мы использовали выше, применяется краткая, с оператором:=. Выражение

```
i := 10
```

эквивалентно

```
var i int = 10
```

```
andrei@ubuntu:~/bin$ ./6.out
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
I'm routine 1
I'm routine 2
$$$
andrei@ubuntu:~/bin$
```

› **Наше первое многопоточное приложение Go.**

Многопоточность

Простая реализация многопоточности объявлена одним из основных преимуществ языка Go, и естественно, что в этом кратком обзоре мы уделим ей основное внимание. В программе, написанной на Go, очень легко сделать так, чтобы несколько процедур выполнялись одновременно. Разработчики Go называют такие процедуры «goroutines» [игра слов с «coroutines», – *прим. ред.*]. Не разделяя их склонности к терминотворчеству, я буду применять термин «со-процедуры».

Знакомство с многопоточностью в Go мы начнем с простого, но поучительного примера:

```
package main
import . "fmt"
import . "time"
func concf(num int) {
var i int = 0
for i < 10 {
Sleep(100)
Printf("I'm routine %d\n", num);
i++
}
}
func main() {
go concf(1)
go concf(2)
Sleep(10000000)
}
```

Начнем по порядку. Помимо пакета fmt, в этой программе нам понадобится пакет time. Обратите внимание, что оба они загружаются с одинаковым псевдонимом – таким, что нам не придется добавлять префикс к именам функций. Этот трюк сработает, естественно, только в том случае, если пакеты не экспортируют одинаковых имен.

Со-процедурой Go может быть любая функция этого языка. В нашем примере это concf(). Функция concf() выполняет цикл for, который синтаксически совсем не похож на одноименные операторы C или Паскаля. На самом деле цикл for в Go является аналогом while, и неудивительно, что последний в Go вообще отсутствует. В цикле функция concf() вызывает функцию Sleep() из пакета time, а затем распечатывает значение своего аргумента. Далее выполняется инкремент управляющей переменной цикла. Обратите внимание на то, что функция Sleep() измеряет интервалы в наносекундах.

В главной функции программы мы дважды вызываем функцию concf(), таким образом, чтобы оба экземпляра выполнялись одновременно. Как вы уже, наверное, поняли, создание нового потока в Go выполняется с помощью конструкции

```
go <имя_функции>(<параметры>)
```

После запуска двух дочерних потоков мы вызываем функцию Sleep() – для того, чтобы главная функция не завершилась до окончания выполнения потоков (при выходе из нее программа завершится независимо от того, работают ли дочерние потоки). В результате мы увидим, как оба потока попеременно распечатывают значения своих параметров.

Стоит отметить, что, по утверждению документации, функция Printf() реентерабельна. Учитывая то внимание, которое в Go уделяется многопоточности, можно ожидать, что все функции из стандартных пакетов обладают таким свойством, однако документация молчит об этом.

Разумеется, использование функции Sleep() в расчете на то, что потоки успеют завершится до истечения заданного интер-

2b ∨ ¬ 2b, вот в чем вопрос

Из-за особенностей типа string мы не можем написать

```
s <- c
```

– нам приходится использовать оператор присваивания; однако, если бы переменная s имела простой тип, например, int, такая запись была бы вполне допустима. Конструкция = <- введет в ступор любого программиста, незнакомого с Go, и, возможно, повысит вашу хакерскую репутацию (хотя обратный эффект тоже не исключается).

вала – не лучший метод управления. Серьезная многопоточная программа не может обойтись без средств синхронизации.

Средства синхронизации

Хотя все, что поддерживает многопоточность в Go, может быть реализовано и в любом другом развитом языке программирования, инструменты Go действительно очень удобны. Разработчики учли, что обмен данными между потоками всегда требует синхронизации, и потому важнейшее средство обмена данными, каналы, является одновременно и синхронизирующим. Рассмотрим простейший пример программы, использующей каналы для передачи данных и синхронизации.

```
package main
import . "fmt"
import . "time"
var c = make(chan string, 1)
func writer() {
Printf("Writer started\n")
Sleep(100000000)
c <- "Hello from writer"
}
func main() {
var s string
go writer()
Printf("Writer is called\n")
s = <- c
Printf("%s\n", s)
}
```

Каналом в нашей программе является переменная **c** типа «канал» (chan). Он относится к сложным типам, поэтому для инициализации **c** используется функция make(), с которой мы еще встретимся далее. Сейчас я просто скажу вам, что мы создаем буферизованный канал для передачи переменных типа string. Функция writer записывает в канал строку (вызов Sleep() добавлен для наглядности). Запись данных в канал и чтение из него выполняются с помощью оператора <-, который чем-то похож на << в C++.

Функция main() вызывает writer() в режиме со-процедуры, а затем читает данные из канала. Важное свойство буферизованных каналов заключается в том, что чтение начинается после того, как заканчивается запись. Таким образом мы гарантируем, что выполнение программы закончится только после того, как writer() запишет данные в канал, а main() прочтет их и распечатает. Строка «Hello from writer» будет последней, которую выведет наша программа. Обратите внимание, что последовательность вывода на печать строк «Writer is called» и «Writer started» непредсказуема, поскольку в этом случае мы не позаботились о синхронизации.

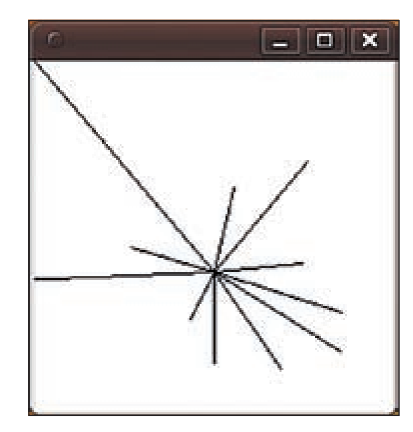
Каналы как средство синхронизации потоков настолько удобны, что их иногда используют исключительно с этой целью (создается буферизованный канал для переменных типа int или byte, а затем запись и чтение значения соответствующего типа применяются в роли сигналов об окончании некоей операции). Тем не менее, каналы – не единственное средство синхронизации потоков в Go.

```
package main
import . "fmt"
import . "sync"
var m1, m2 Mutex
var i int = 0
var c = make(chan int, 1)
func finish() {
c <- 1;
}
func join(num int) {
for i < num {
i = i + <- c
}
}
func printer1() {
m1.Lock();
Printf("This is ")
m2.Unlock()
m1.Lock()
Printf("that Jack ")
m2.Unlock()
finish()
}
func printer2() {
m2.Lock();
Printf("the house ")
m1.Unlock()
m2.Lock()
Printf("built\n")
m1.Unlock()
finish()
}
func main() {
m2.Lock()
go printer2()
go printer1()
join(2)
}
```

Эта программа использует два потока и мьютексы для вывода на печать строки из детского стишка. Объект Mutex определен в пакете sync. Для тех, кто не знает, как работает мьютекс, упрощенное объяснение: если поток А вызывает метод m.Lock() мьютекса m, а затем поток Б также вызывает метод m.Lock(), выполнение Б будет приостановлено до тех пор, пока А не вызовет m.Unlock(). Введенные мной функции finish() и join() позволяют программе дождаться завершения всех со-процедур. Перед выходом из со-процедуры мы вызываем функцию finish(), а вызов функции join(n) приостанавливает выполнение вызвавшей ее процедуры до тех пор, пока n со-процедур не вызовут finish(). Функции finish() и join() используют буферизованные каналы в роли семафоров (как это часто делается в Go). Как ни странно, я не нашел аналога этих удобных функций в документации по Go, так что прошу рассматривать их как мой собственный вклад в развитие среды времени выполнения этого языка.

На закуску предлагаю вам самостоятельно разобрать текст программы на Go, которая использует графический интерфейс X11 (рис. 4). Вы найдете его на диске в файле **xwindows.go** – это один из примеров от авторов Go, слегка модифицированный мной. Результат его работы (причем в окне Microsoft Windows!) можно видеть на рисунке. LXF

› На Go можно писать и графические приложения.



Взаимные блокировки

Экспериментируя с со-процедурами, я обнаружил интересную вещь. Если нарочно заблокировать с помощью мьютексов выполнение всех со-процедур, программа не повиснет, а завершится аварийно, с сообщением "throw: all goroutines are asleep - deadlock"! Наличие «сторожевой собаки», проверяющей взаимные блокировки, безусловно, интересная и полезная функция Go, но ее возможности не следует преувеличивать. Как известно, невозможно написать программу, способную проверить правильность любой другой: «умелые руки» всегда найдут способ поставить в тупик искусственный интеллект.

Что за штука... MicroXwin?

Новая оконная система бальзамом оросит душу разработчиков встраиваемого Linux. **Марко Фиоретти** объяснит, почему...

Так что же за штука *MicroXwin*?
MicroXwin (www.microxwin.com) – замена системе *X Window System*, обычно применяемой в Linux и Unix.

Хорошо, а что за штука *X Window System*?
X Window System, или просто *X* – стандартная система, используемая всеми приложениями Linux и Unix для взаимодействия с графическими драйверами с целью отображения объектов, операций с окнами, контроля движений мыши и т.д. Особую гибкость системе *X* придают два свойства: архитектура клиент–сервер и сетевая прозрачность.

Что означает «архитектура клиент-сервер»?
Используя ПО с графическим интерфейсом (GUI) на основе *X*, вы всегда имеете дело с двумя совершенно разными программами: *X*-сервером и *X*-клиентом. Сервер контролирует графический чипсет компьютера и отображает изображение на мониторе. Клиенты – это прикладные программы вроде *OpenOffice.org*, *GIMP* или *Firefox*. Если вы предпишете им открыть окно или нарисовать меню, они обратятся с соответствующим «предложением» к *X*-серверу, как к распорядителю ресурсов. При нажатии кнопки или перемещении мыши сервер сообщает об этом клиентам, чтобы они могли реагировать на ваши действия.

> «Графическая система впервые интегрирована в ядро как модуль.»

А как сервер общается с разными клиентами?
Как правило, *X*-клиенты пишут с помощью высокоуровневых графических инструментариев, типа *GTK* или *Qt*. А с сервером они связываются по протоколу *X Window System*, применяемому в низкоуровневых программных библиотеках под названием *Xlib*.

Осталось объяснить, что такое сетевая прозрачность...
Это означает, что *X*-клиент – то есть используемое вами ПО – может работать вовсе не на том компьютере, за которым вы сидите, а вы даже не заметите разницы. *X*-сервер на вашем компьютере будет взаимодействовать с удалённой программой тем же способом, что и с локальной. Именно сетевая прозрачность делает такие проекты, как *Linux Terminal Server*, полезными и простыми в настройке.

Зачем тогда нужен *MicroXwin*? Чего нам не хватает в *X*?
При обычной работе – всего хватает. Однако в ряде других ситуаций данное решение слишком сложно: «виноваты» те самые функции, что придают ему удобство и гибкость. Сетевая прозрачность и протокол *X* сильно «утяжеляют» сообщения, пересылаемые между сервером и клиентами. Многоступенчатый процесс пожирает уйму процессорного времени, даже если клиент и сервер выполняются на одной машине. В последнем случае дело осложняется непрерывным переключением контекста: ядро постоянно «прыгает» между клиентами и сервером (это необходимо для взаимодействия между ними).

Ну и что? Процессоры в наше время мощные, память дешева...
Для настольных компьютеров – да. Но ведь Linux работает в бесчисленных встраиваемых устройствах, от мобильных телефонов и автомобилей до домашней электроники. Процессоры таких приборов (например, ARM 9) заметно «тормозят», выполняя все формальности *X Window System* (а порой и неспособны её вытянуть). Система *MicroXwin* разработана специально для таких случаев, хотя использовать её на настольном ПК тоже можно.

А в чем главная внутренняя разница *MicroXwin* и *X*?
То, что *MicroXwin* отказывается от описанных выше двух свойств, а значит, будет гораздо быстрее и компактнее обычного *X*-сервера.

За счет чего?
MicroXwin – первая графическая система, которая может интегрироваться в ядро Unix/Linux как модуль. Она копирует данные прямо из пользовательского пространства в буфер кадров, а оттуда – на экран. Это полностью устраняет проблему переключения контекстов и позволяет начать вывод графики на более ранних стадиях загрузочного процесса.

Есть ли другие компоненты?
Кроме собственно ядерного модуля, есть ещё две главные библиотеки. Одна их них служит для отправки графических команд на модуль и будет заменой *Xlib* (720 КБ). Вторая библиотека является основой для написания расширений *MicroXwin* (7 КБ). Модуль ядра для платформы x86 «весит» около 300 КБ.

Велик ли выигрыш в скорости?
Сайт *MicroXwin* сообщает о значительном преимуществе в скорости по сравнению с *X.org* (стандартным *X*-сервером большинства современных дистрибутивов) по результатам тестов *Gtkperf* и *X11perf*. Все подробности – на сайте.

И сколько памяти требует *MicroXwin*?
Для модуля необходимо около 500 КБ памяти ядра, против 18–20 МБ, используемых (неработающим) сервером *X*.

Невероятно! И всё это совместимо со стандартными *X*-клиентами?
Да – ведь интерфейс прежний! Разработчики добиваются, чтобы программы могли работать на *MicroXwin* после простой перекомпиляции или предписания использовать библиотеки *MicroXwin* вместо традиционных *Xlib*. К настоящему времени цель почти достигнута.

Ну, а стабильность? Тесная интеграция между графикой и ядром Windows – одна из причин появления «синих экранов смерти»...
Хотя и нечасто, у традиционных *X*-серверов тоже случаются сбои, об этом любой Linux-пользователь знает. Что касается *MicroXwin*, то разработчики заметили, что ошибки в любом драйвере

или модуле ядра способны привести к зависанию или краху Linux.

То есть, если система с *MicroXwin* «падает»...
...значит, в модуль данной версии вкралась ошибка; такое бывает и в *X*. Тут нет связи с тем, что это модуль ядра, а не сервер пользовательского пространства.

***MicroXwin* работает только на Linux?**
Согласно сайту, новую систему легко портировать на любую платформу.

Могу ли я написать клиент *MicroXwin*, используя мой любимый инструментарий или оконный менеджер?
Конечно, потому-то и борются за совместимость со стандартным интерфейсом *Xlib*. Чтобы ваши приложения работали с *MicroXwin*, достаточно иметь библиотеки и модуль ядра.

Секундочку: скажите, а это всё свободное ПО?
Библиотеки – да, а вот модуль ядра проприетарный.

Ну вот, модуль ядра — и проприетарный...
Такой вопрос задавали на сайте *MicroXwin*. Пересказываю своими словами: чтобы программы были совместимы с *MicroXwin* и оставались в сфере открытого ПО, разработчикам FOSS необходимы только свободные библиотеки. Программисты *MicroXwin* объяснили, что деньги от лицензирования модуля ядра нужны им для продвижения разработки *MicroXwin* на системах Linux и Unix. Если задуматься, аргументы убедительные, и всё это не должно негативно сказаться на развитии свободного ПО.

Почему?
Как мы уже упоминали, современным ПК и ноутбукам хватает мощности, чтобы обойтись без *MicroXwin*. Модуль ядра *MicroXwin*, лицензированный на условиях GPL, пригодился бы разве что для оживления списанных компьютеров x86, но платить за это никто не будет. Основные рынки *MicroXwin* – где новая система будет полезной, доходной и привлекательной – это смартфоны, мобильные интернет-устройства, нетбуки и прочие встраиваемые системы. Но большинство этих платформ в любом случае закрыты, то есть модифицировать или устанавливать соответствующие ОС имеют право только производители устройств.

То есть лицензия ядерного модуля не коснется конечного потребителя и разработчика?
Да, верно. Но она очень даже касается производителя аппаратного обеспечения, желающего расширить сбыт своей продукции. Покупка лицензии *MicroXwin* тут же делает его разработку совместимой с тысячами готовых Linux-приложений. Поэтому в долгосрочной перспективе можно продать значительно больше устройств, чем без лицензии.

Так что же год грядущий готовит *MicroXwin*? Куда движется проект?
Версия *MicroXwin 2.0*, включающая обновленную поддержку *XRender*, была выпущена в январе. Кроме того, разработчики планируют закончить в 2010 году портирование на микропроцессоры ARM. Это чипы для нетбуков и мобильных интернет-устройств, выпуск которых на потребительский рынок ожидается в текущем году. LXF





По рецептам доктора Брауна

Д-р Крис Браун
Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Добротное администрирование систем из причудливых заворотов кишок серверной.















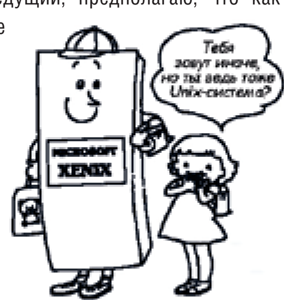

Microsoft поставляет Linux...

Такой анонс мы вряд ли скоро увидим. Но это менее невероятно, чем кажется. Microsoft, пожалуй, *начнет* поставлять Linux, если на то появятся бизнес-причины. Например, если она решит, что незачем продолжать разрабатывать собственную операционную систему, когда есть прекрасное ядро, написанное кем-то другим.

Конечно, это будет «MS Linux», далекий от того Linux, что мы знаем, с неясными и недокументированными изменениями в ключевых системных вызовах (просто чтобы добавить экстрима к портированию на него существующего открытого ПО), и, без сомнения, проприетарным форматом пакетов, который всему остальному миру придется вскрывать. Но если будут портированы *Internet Explorer*, *Office* и рабочий стол, для пользователя эта система будет мало чем отличаться от Windows.

История Unix

Невероятно? Пожалуй. Но вспомним, что Apple отказалась от своего проприетарного ядра ради ядра Darwin, которое по крайней мере частично основано на BSD Unix. А Microsoft серьезно пофлиртовала с Unix еще в начале 1980-х – и плодом была продажа продукта под названием Xenix. Позже она, конечно, выпустила несколько драйверов для ядра Linux под GPL2, выступив в роли настоящего разработчика Linux. Как человек совсем несведущий, предполагаю, что как минимум двое из Microsoft об этом говорили, но вряд ли в радиусе слышимости мистера Балмера.



› **Свидетельство недолгого романа Microsoft и Unix. Подробности на сайте www.tenox.tc/tmp/ww/msxenix/pics.**

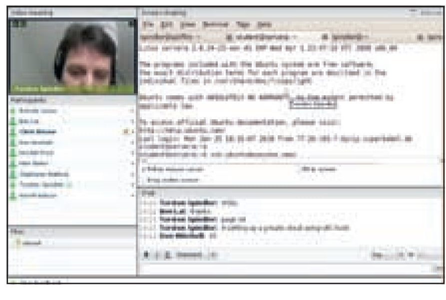
Виртуальные классы

Ведомые педагогом Крис делится опытом работы с двумя технологиями-соперницами.

Недавно я посещал учебный курс Canonical – он проводился в виртуальной классной комнате с помощью *Spreed*, сервиса для организации онлайн-конференций. Неделей раньше я стоял «по ту сторону доски» с другой подобной технологией, *Adobe Acrobat Connect Pro*, проводя в Learning Tree курсы одновременно для настоящих и удаленных слушателей.

Эти продукты похожи. И тот, и другой – web-сервис, работающий через браузер. Обоим необходим Flash, оба поддерживают живые видеопотоки, содержат окно чата для студентов, список участников конференции для обратной связи, область для представления контента или разделения монитора преподавателя. В продукте от Adobe можно настроить расположение элементов на экране. Благодаря этому Learning Tree поддерживает свой фирменный стиль презентации с двух проекторов. Оба продукта представляют двустороннее управление аудиопотоком, причем хозяин конференции может включать и выключать микрофоны отдельных участников.

Заставить продукт от Adobe работать в *Firefox* и Linux я не смог; требуется модуль расширения, который не прилагается. Spreed хорошо работает в тех же условиях с последней версией модуля для воспроизведения *Flash*. Две компании используют разные технологии для поддержки удаленного входа на компьютеры для практических упражнений. Learning Tree применяет проприетарный протокол удаленного рабочего стола, которому нужен модуль расширения Active-X для браузера, а Canonical – более простое сочетание SSH и перенаправления локальных портов.



› **В *Spreed* и *Acrobat Connector* есть множество способов взаимодействия с преподавателем.**

В этих системах мне нравятся короткие конференции, где я являюсь активным участником; но при длинных переговорах, когда моя роль более пассивна, я уже с трудом концентрируюсь.

Сколько стоит Spreed?

На *Spreed* у вас будет бесплатная учетная запись, позволяющая проводить конференции длительностью до 90 минут с числом участников до трех. Подписка Channel-20 стоит примерно 4300 рублей в год и предоставляет виртуальную комнату для конференций, где могут поместиться до 20 участников. Чтобы проводить две конференции одновременно, нужно купить две подписки на канал. Можно купить комнату и «на одноразовую конференцию»: например, Webinar-10 представляет 10-часовую конференцию для 10 участников за 860 рублей. Подробности – на сайте http://spreed.com/prices/overview.

О процессах ясно

Объясняем жаргон О процессах слышали все, но что, собственно, это такое?

Нас пришла навестить тетушка Мод – джин пришлось припрятать. Тетя Мод что-то слышала о мире ИТ, но знала, что я «эксперт», и (к ее чести) хотела узнать еще. Она спросила: «А где программы, которые работают на моем компьютере? Если мы заглянем внутрь, сможешь их показать?» На этот вопрос было непросто ответить так, чтобы тетя Мод поняла. Конечно, можно было сказать, что программа – это «последовательность команд», и, указав на жесткий диск или чипы памяти, заявить: «Вот тут они и живут».

Мы спрятали и виски, потому что ее муж, Клайв, тоже пришел. Его познания простирались несколько дальше, и потому он был еще опаснее тети. Он спросил: «Если Linux – многопроцессная операционная система, то где же процессы? Если открыть компьютер, сможешь их показать?» Этот вопрос посложнее, потому что по сравнению с программой процесс – более абстрактное понятие. Я мог бы выдать готовый ответ, сказав, что «процесс – это экземпляр выполняемой программы», но это вряд ли бы помогло бедному Клайву. Да и мне бы не помогло.

Нудные основы

Итак, вот мое объяснение, что же такое процесс. Оно ни сжато, ни красочно – я не думаю, что так вы разберетесь в этой штуке. Программе, выполняемой в Linux, нужны некие ресурсы. Ей требуется память (кодовый сегмент, где хранится она сама, сегмент статических данных, стек и «куча», где содержатся данные). Программа может открывать файлы или подключаться к определенному серверу. Эти соединения отслеживаются с помощью «файловых дескрипторов». У программы есть понятие о «текущем каталоге», где она будет искать файлы, если их имена начинаются не с '/'. У нее есть «окружение» с «переменными окружения», вроде **HOME=/home/chris**. Наконец, программе нужно процессорное время – действительная возможность быть запланированной на выполнение в соперничестве с другими процессами, требующими того же. Именно процесс хранит весь контекст и ресурсы, необходимые для запуска программы.

> «Процесс хранит контекст и ресурсы для запуска программы.»

fork ( )

Родительский процесс запускает копию оригинальной программы

Родительский процесс

exec ( )

Дочерний процесс запускает новую программу

wait ( )

Код завершения передается назад родителю

› Типичный жизненный цикл процесса в виде системных вызовов родительского и дочернего процессов.

```
$ pstree
init-+-BackupPC---BackupPC_trashC
     |-NetworkManager-+-dhclient
     |                `-{NetworkManager}
     |-gnome-terminal-+-bash
                      |-bash---pstree
                      |-gnome-pty-helpe
                      `-{gnome-terminal}
```

› Процесс *init* – родитель терминала Gnome, который запускает оболочку *Bash*, которая запускает *pstree*.

Здесь может помочь аналогия с актером, читающим сценарий пьесы. Сценарий – набор инструкций, поясняющий актеру, что говорить и как вести себя на сцене. Сценарий можно сравнить с программой. Актер – объект, ответственный за чтение сценария, и подобие процесса.

Новый процесс создается тогда, когда в существующем процессе происходит системный вызов **fork()**. Исходный процесс называется родительским, а новый – дочерним. Сначала дочерний процесс представляет собой точную копию родительского и выполняет ту же самую программу. В моем примере с актерами, два актера теперь стоят рядом и читают один и тот же сценарий. Иногда дочерний процесс продолжает выполнять ту же программу, что и родительский: например, когда *Apache* порождает дочерние процессы, все они выполняют один и тот же код. Но чаще дочерний процесс перестает работать и запускает другую программу с помощью системного вызова **exec()**. В моем примере актер видит указание отложить текущую пьесу, взять текст «*Макбета*» и начать сначала.

Процесс завершается системным вызовом **exit()**. Его можно прервать и с помощью сигнала: об этом мы поговорим через месяц. Когда дочерний процесс прекращает работу, он возвращает код завершения (exit status) родительскому процессу, для сообщения о том, что все закончилось хорошо (код завершения **0**) или что-то пошло не так (значение, отличное от нуля). Родительский процесс может ждать, пока завершится дочерний, и получить его статус с помощью системного вызова **wait()**. Такое происходит, когда вы набираете команду в оболочке: она создает для переданной команды новый процесс и ждет, пока он завершится.

Но родительский процесс не обязан дожидаться дочерних. Когда из меню рабочего стола запускается графическое приложение или оболочкой запускается фоновое задание, родительский процесс просто продолжает работать вместе с дочерним.

Просмотреть процессы, выполняющиеся на компьютере, можно командой **ps**. У нее есть много флагов для выбора отображаемых процессов и информации о них, но особенно удобно начать с **ps -ef**. Команда произвела глубокое впечатление на моего дядюшку Клайва.

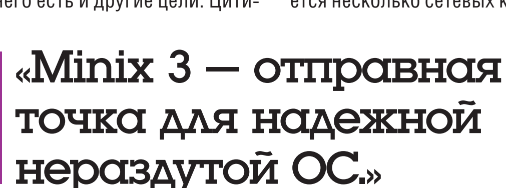
Мини-Linux

Minix OS Без Minix, наверное, не было бы и Linux. Рассмотрим же его.



Линус Торвальдс писал исходное ядро Linux под впечатлением от другой Unix-подобной системы, Minix, созданной Эндрю Таненбаумом [Andrew Tanenbaum]. Торвальдсу был знаком Minix, поскольку он первоначально пользовался им как платформой для разработки ядра Linux. И, несомненно, на него повлияла книга Таненбаума «*Операционные системы: разработка и реализация*»: Торвальдс называл ее «книгой, которая подняла меня на новые высоты». Нельзя сказать, что архитектура Minix нравилась ему на 100 % – вообще-то он относился к ней довольно критически. Почитайте, например, книгу Линуса Торвальдса и Дэвида Даймонда [David Diamond] «Just for FUN. Рассказ нечаянного революционера» или споры Торвальдса и Таненбаума по адресу http://bit.ly/KmLnQ. Minix создавался в основном как средство для обучения. Еще в 1987 году Таненбаум (профессор факультета вычислительной техники университета Врейе в Амстердаме) хотел создать операционную систему, которая бы иллюстрировала принципы его книги и которой могли бы пользоваться его студенты.

Довольно долго я имел об этом слабое понятие, пока случайный поиск в Google не привел меня на www.minix3.org. Я и не подозревал, что Minix все еще в активной разработке. Оказывается, в 2008 году Таненбаум получил грант в 2,5 млн евро на продолжение развития Minix, и последний релиз вышел в ноябре 2009 года. Minix 3 продолжает играть роль ОС, которую студенты могут расчленить на составляющие, но у него есть и другие цели. Цитирую сайт: «Он примерно основан на предыдущих версиях Minix, но отличается по многим ключевым вопросам. Minix 1 и Minix 2 создавались как средства для обучения, Minix 3 намерен стать полноценной системой для компьютеров с ограниченными ресурсами и встраиваемых, а также для приложений, требующих высокой надежности».



Где-то в другом месте Таненбаум написал: «Minix 3 – это не Minix вашего дедушки. Minix 1 создавался как средство для обучения и все еще широко используется в университетах как таковое. Minix 3 – то же самое плюс отправная точка построения высоконадежной самовосстанавливающейся нераздутой ОС, наверное, подходящей для таких проектов, как стодолларовый ноутбук в помощь детям стран третьего мира или встраиваемая система. Minix 1 и Minix 3 объединяет то же, что и Windows 3.1 и Windows XP – первое слово в названии».

Кто хочет сам поиграть с Minix, загрузите сжатый ISO-образ с сайта www.minix3.org/download. Он распространяется по BSD-подобной лицензии. CD работает в режиме Live, и вы получите определенное представление о легкости Minix, когда увидите, что командная строка загрузится менее чем за три секунды. После старта системы нетрудно установить ее на жесткий диск.

Низкие требования

Требования к аппаратной части минимальны: стандартная система работает на 16 МБ ОЗУ, а на 1 ГБ дискового пространства умещается весь дистрибутив, включая исходные коды. Поддерживается несколько сетевых карт, но на данный момент нет поддержки Wi-Fi, USB и FireWire. Я установил ее на виртуальную машину *VMWare*, и она прекрасно работала.

Загрузка Minix – возвращение в более спокойные, простые и в общем более интересные времена начала эпохи Linux и Unix, когда вся схема легко укладывалась в голове. А может, моя голова тогда была больше. Пользователям Linux Minix покажется довольно знакомым, по крайней мере в командной строке: это POSIX-совместимая ОС, и в нее были портированы многие классические утилиты – *ls*, *Cron*, *GCC*, *Perl*, *SSH*, *file*, *find*, *sort*, *mount*, *Vi*, *grep* и прочие. В **/usr/bin** в общей сложности 338 команд, и многие вы узнаете (сравните с 2048 в моем Ubuntu). Архитектура файловой системы Minix тоже покажется вам весьма знакомой.

А под капотом между Minix и Linux имеются заметные отличия. Например, в Minix единственный дисковый раздел делится на «слайсы», и им даются имена по схеме, похожей на принятую в Solaris. Например, первый диск на первом контроллере будет **/dev/c0d0**, а первый раздел на этом диске – **/dev/c0d0p0**. В терминах Linux это были бы **/dev/sda** и **/dev/sda1** соответственно. Слайсы именуются как **/dev/c0d0p0s1**. В Linux нет концепции слайсов, применяемой в Minix.

Более существенное различие между Minix и Linux в том, что архитектура первой построена на основе микроядра. Само ядро предоставляет обработчики прерываний, способ запуска и остановки процессов, планировщик, основы межпроцессной коммуникации, но ничего более. Большая часть функционала, который в Linux находится в ядре, реализована в Minix в пространстве пользователя. Это драйверы устройств, файловая система, стек

Типы ядра

Термин «монолитное ядро» означает две разные вещи, и по этому поводу у меня однажды была запальчивая дискуссия с сертифицированным архитектором Red Hat. В контексте сравнения Minix с Linux монолитное ядро означает, что вся операционная система выполняется в ядре, в отличие от микроядра. Но в последние годы в мире Linux термин «монолитное ядро» начал означать, что весь код ядра статически скомпонован в один исполняемый образ. Такая схема отличается от модульного ядра, многие возможности которого реализованы в динамических модулях, подгружаемых по требованию. В этом смысле слова ранние версии ядра Linux были монолитными, а все последние версии – модульные. Динамическая загрузка модуля, реализующего файловую систему в пространстве ядра (как в Linux), отличается от ее реализации в виде сервиса в пространстве пользователя в Minix.

Что входит в ядро?

Коды, выполняемые «в привилегированном режиме» (ядро) и «в пространстве пользователя», серьезно различаются. Но что относится к ядру, и что – к пространству пользователя? Есть определенный минимум вещей, которые должны выполняться в привилегированном режиме: низкоуровневое управление памятью, обработка прерываний, планирование процессов и некое минимальное межпроцессное взаимодействие [IPC]. На другом конце шкалы, по общему мнению, находятся пользовательские приложения типа *Bash* и *Firefox* – они принадлежат к пространству пользователя. Однако между ними есть «серые зоны». Как насчет библиотеки математических функций, сервера NFS, стека TCP/IP, кода управления доступом, драйверов устройств или файловой системы ext3?

В истории разработки Linux вы найдете примеры, когда те или иные вещи пересекали границу между пространством пользователя и пространством ядра. NFS изначально была демоном в пространстве пользователя, а потом переместилась в ядро. С другой стороны, относительно новая технология Fuse позволяет реализовать в пространстве пользователя файловые системы, обычно принадлежащие ядру.

Архитектуре микроядра посвящен большой объем исследовательской литературы, который вы вряд ли захотите осваивать, если только не пишете главу об истории архитектуры операционной системы в своей диссертации. Тогда загляните на сайт http://bit.ly/4xemNx.

протоколов TCP/IP и даже высокоуровневое управление памятью. В пространстве пользователя вы увидите процессы для VFS (виртуальной файловой системы), TTY (драйвера терминала) и DS (сервера хранилища данных). Это фундаментальное различие в архитектуре, и здесь взгляды Торвальдса и Таненбаума определенно не сходятся (см. врезки «*Типы ядра*» и «*Что входит в ядро?*»). Чтобы вы представили себе масштаб этого различия, скажу, что в ядре Minix 3 около 4000 строк кода, а в ядре Linux 2.6.32 – около 12 млн. Пока вы не начали возмущаться, соглашусь, что это не совсем корректное сравнение размера двух ОС. Но оно иллюстрирует компактность ядра Minix.

Упакуйте это

В Minix есть примитивный менеджер пакетов *Packman*. Это сценарий оболочки, и его пользовательский интерфейс довольно громоздок. Зависимостей у него нет, и устанавливается пакет простым копированием архива с CD (или с сайта Minix www.minix3.org/packages/i386) и его распаковкой. Архивы с исходным кодом также доступны по адресу www.minix3.org/software, и *Packman* спрашивает, хотите ли вы их установить.

Также есть работающий *X*-сервер и два оконных менеджера. Первый из них, *Twm*, сокращение от *Tab Window Manager*, появился в 1987 году и по сегодняшним меркам довольно примитивен, но в свое время был прорывом. Второй, *Ede* (*Equinox Desktop Environment*), выглядит чуть современнее, но все так же легок и быстр. Не ждите от Minix последних новшеств – если вы гонитесь за этим, вы попали не туда. Minix далеко до всеохватности и элегантности Linux, но он и не пытается таким быть.

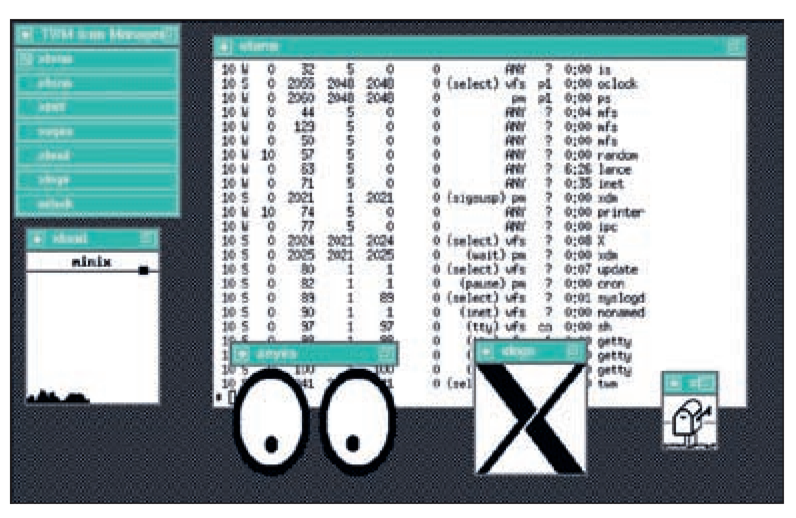
Итак, здесь меньше утилит, система управления пакетами примитивна, рабочий стол – древний; в чем же привлекательность Minix? Ну, для разработчика он гораздо проще и доступнее Linux. Например, для повторной сборки и установки операционной системы и всех утилит достаточно двух команд:

```
#cd /usr/src
#make world
```

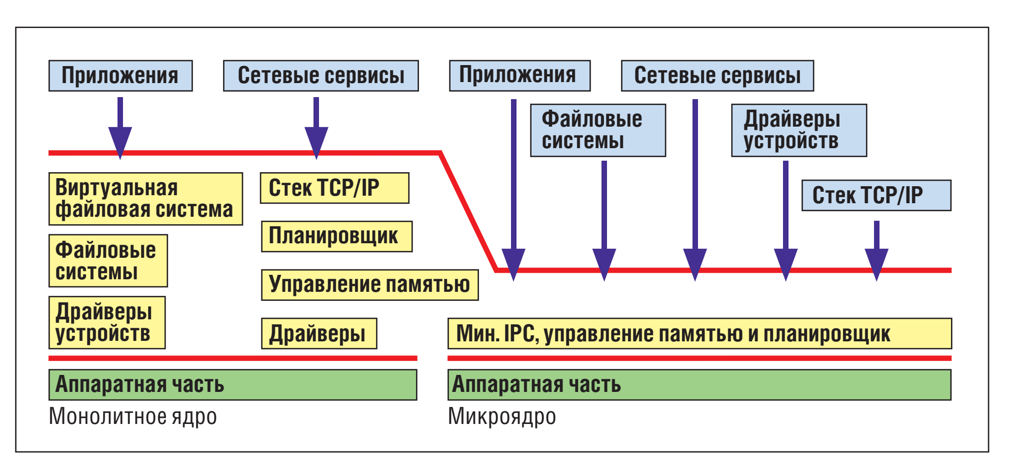
В нашей тестовой системе это заняло минуты три. Не представляю, сколько тут ушло бы времени в Linux, но подозреваю, что займись этим Моцарт, он бы без труда написал пару симфоний, дожидаясь завершения.

Но шутки в сторону: Minix допускает анализ исходного кода реальной операционной системы без вывихивания мозгов. И если вас по-настоящему вдохновили его возможности, взгляните на несколько проектов на wiki.minix3.org/en/StudentProjects и прикиньте, как будет смотреться ваша фамилия с приставкой «к. т. н.» или «д-р».

Загружая Minix, я довольно внезапно ощутил беспомощность, сообразив, что не понимаю, как сделать многие простые вещи. Я не знал, как изменить IP-адрес моего сетевого интерфейса или как назначить сервисы, стартующие во время загрузки. Это напомнило мне неуверенность, которую я испытывал в первые дни работы с Linux, и послужило уместным уроком того, что к новичкам в Linux нужно относиться с огромным терпением и уважением. LXF



› **Minix с оконным менеджером *Twm*, *Xterm*, программой для оповещения о новых письмах и *Xeyes*. По-вашему, он выглядит примитивно? Зато работает на 8 МБ ОЗУ.**



› **Микроядро, как в Minix, выносит как можно большую функциональность в пространство пользователя и дает более компактное ядро. Объем его кода в Minix всего 4000 строк.**

Учебники

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Не страшная ОС

Пугать детей консолью не надо.
Я пытался – не испугались.
Енот Васильевич

Компьютер – помощник человеку, и негоже быть человеческим интерфейсом с дискеткой в руке при умной машине. Ему нужно давать команды, и для этого окружение GNU/Linux предоставляет массу возможностей. Да, глаза разбегаются от них, но многообразие не страшно, если из него можно свободно выбрать, а выбрав, использовать оное, не боясь внезапной смены поведения.

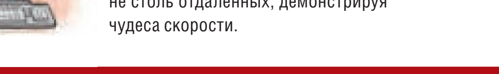
Ярким примером может быть *tar*, чью очередную версию за номером 1.23 недавно отпраздновали истинные энтузиасты резервного копирования. Столько лет прошло с его создания, а он стабильно спасает в простейших и изощрённых ситуациях. Конкурентов море, но *tar* и не думает меняться, так как в своей нише архивной рабочей лошадки он идеален.

Свободные программы не умирают и не перерождаются, а только аккумулируют всё в себе. На кратком промежутке времени качественные проприетарные продукты могут успешно конкурировать со свободными аналогами, но что будет, скажем, через двадцать лет? То, что будет интересно – это бесспорно.

P.S. Компас-3D с помощью WINE@Etersoft CAD теперь может официально работать в GNU/Linux. Видимо, тоже уже не боятся.

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...





Совет месяца: Двоичная магия

Исполняемые файлы (иными словами, программы, состоящие из двоичного кода), могут выглядеть магией для непосвященных, но в Linux есть несколько инструментов командной строки, позволяющих пролить свет на их суть. Первая из них, **file**, сообщает общую информацию. Например, набрав **file /usr/bin/gedit**, вы увидите

```
/usr/bin/gedit: ELF 32-bit LSB executable,
Intel 80386, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux
2.6.15, stripped
```

Здесь указана платформа, для которой скомпилирован исполняемый файл, а также отмечено, что он использует разделяемые библиотеки. Если **file** сообщает, что интересующая вас программа скомпонована статически (statically linked), значит, все необходимое уже включено в исполняемый образ, и (теоретически) он должен работать, где угодно. Программы с динамической компоновкой (например, наш *Gedit*) ожидают найти требуемые библиотеки в определенных местах файловой системы, перечисленных в **/etc/ld.so.conf**. Узнать, какие динамические библиотеки используются программой, позволит **ldd**:

```
ldd /usr/bin/gedit
```

Вы получите список (в данном случае – достаточно длинный) библиотек, требуемых *Gedit*, и соответствующих им конкретных файлов в вашей системе. Узнать подробности можно в **man ldd**.

CLI: Работа

Не бойтесь командной строки: наш эксперт **Энди Ченнел** поможет побороть страх перед текстовым режимом и покажет вам всю его мощь.





Наш эксперт

Энди Ченнел
Энди, видимо, будет делать Первые шаги в Linux вечно, а технологиями он заинтересовался, открыв для себя Dragon 32.

Вероятно, в прошлом Linux более всего пугал необходимостью иногда работать тет-а-тет с огромным пустым экраном, располагая лишь парой волшебных слов, способных помочь прояснить ситуацию. Для аллергиков на старомодную манеру общения с компьютером графический интерфейс пользователя (Graphical User Interface, GUI) стал даром божьим. Но и сейчас есть шансы оказаться наедине с интерфейсом командной строки (Command Line Interface, CLI), и уж лучше быть к этому готовым.

Просмотр файловой системы в графическом режиме покажет вам, что она организована как древовидная структура: то есть имеется корневая папка, а ниже уровнем располагаются другие, и в них, в свою очередь, также могут быть вложены папки или файлы. Простоты ради, откройте окно терминала – соответствующий пункт вы найдете в меню Приложения > Служебные [Applications > Accessories]; по умолчанию, эта команда выведет вас в вашу папку **/home/имя_пользователя**, и вы увидите приглашение командной строки, нечто вроде

```
user@host ~$
```

Оно содержит имя пользователя (user) и имя компьютера (host), а **~$** отмечает, что вы вошли в систему как обычный пользователь. Ваше точное местоположение в файловой системе покажет команда **pwd**.

Первым делом вам захочется просмотреть содержимое текущего каталога – для этого введите команду **ls**. Вы увидите список папок и файлов, обычно оформленный как таблица. Действие команды **ls** можно немного изменить, приписав к ней «аргумент», обозначаемый символом дефиса (-). Скажем, команда **ls -l** тоже отобразит содержимое каталога, но добавит в вывод справочные данные, включая сведения о размере файла и правах доступа к нему, а также имя его владельца. Часто файлы и папки отображаются по-разному. Например, на экранном снимке в верхнем углу папки отображаются голубым цветом, а файлы – черным.

› **Команда ls -l отображает сведения о ваших файлах; как видите, их цвет зависит от типа.**

Порядок отображения содержимого каталога можно изменить – в *Nautilus* вы бы сделали это щелчком мыши по заголовку столбца. Например, команда **ls -r** отобразит содержимое папки, отсортированное в обратном алфавитном порядке. Аргументы можно комбинировать: так, команда **ls -l -r** (или даже **ls -lr**) выведет подробную информацию о ваших файлах, причем список будет отсортирован в обратном алфавитном порядке.

Гуляем по файловой системе

Просмотрев содержимое папки, можно переместиться в одну из вложенных в нее, посредством команды **cd**. Например,

```
cd Documents
```

переместит вас в вашу папку **Documents**, а ее содержимое опять-таки покажет команда **ls**. Подобным образом вы можете сновать по файловой системе, пока не выйдете на нужные файл или папку. Чтобы подняться в этой иерархии на уровень вверх, наберите **cd ..**.

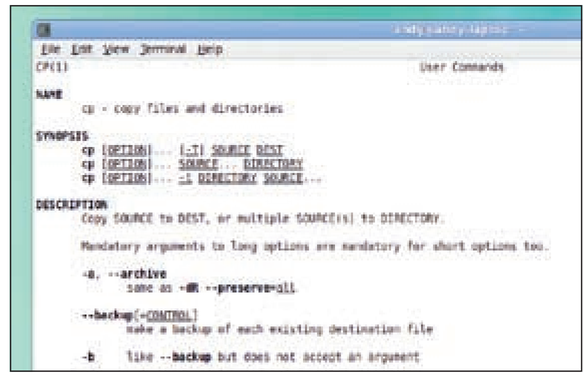
Желая переместиться в совершенно другое место файловой системы, укажите полный, он же – абсолютный, путь к нужной папке:

```
cd /home/andy/Documents
```

Теперь, научившись эффективной навигации по файловой системе, рассмотрим доступные средства для управления файлами. Сперва, однако, убедитесь, что вы находитесь в каталоге (то есть папке), в котором вы имеете права чтения и записи. Для начала создадим новую папку и скопируем туда один или несколь-

» **Месяц назад** Мы редактировали файлы настройки и приручали *Dolphin*.

с командами



› Все опции, доступные в конкретной команде, покажет команда man [имя команды].

ко файлов. Чтобы создать пустой каталог, вложенный в текущий, достаточно ввести следующую команду:

```
mkdir folder1
```

Она создаст новую папку с именем **folder1**. Удаление каталогов производится примерно так же – наберите

```
rmdir folder1
```

Это должно сработать, но только если папка пуста; в противном случае вам сообщат, что операцию выполнить нельзя. Чтобы уничтожить папку со всем ее содержимым, скомандуйте

```
rm -r folder1
```

В ответ на команду удаления, системы с графическим интерфейсом выводят сообщения и создают прочие препоны для опрометчивых; а вот командная строка грохнет все, что ни укажи. Спокойнее, наверное, вставить добавочный аргумент:

```
rm -ir folder1
```

i означает 'interactive'; с ним система потребует подтвердить удаление для каждого файла.

Время от времени бывает нужно перенести каталог с места на место. Тут понадобится строка чуть подлиннее – например, команда

```
cp -r folder1 folder2
```

создаст точную копию папки **folder1** и присвоит ей имя **folder2**. Аргумент **-r** важен: он гарантирует, что папка скопируется вместе с содержимым. Ну, а если вы запустите эту команду, когда у вас уже имеется каталог с именем **folder2**, то папка **folder1** скопируется внутрь него, сохранив исходное имя. Путь к ней будет **/home/user/folder2/folder1/**.

Для перемещения каталога используется команда **mv**, аналогичная **cp**. Так, команда

```
mv folder1 folder2
```

переименует папку **folder1** в **folder2** – это если папки **folder2** у вас нет. В противном случае команда переместит папку **folder1** вовнутрь **folder2**.

Заметьте, что и команда **cp**, и команда **mv** работают и с относительными, и с абсолютными путями, и можно скопировать папку из домашнего каталога одного пользователя в домашний каталог другого (при условии, что у вас есть права доступа к обоим) командой

```
cp /home/rita/secrets /home/bob/ritas_secrets
```

В строке указаны команда, а также исходная и целевая папки с абсолютными путями к ним.

Эти две команды точно так же работают и с файлами, надо просто приписать их имена к полным путям исходной и целевой папки:

```
cp /home/rita/secrets/diary.odt /home/bob/ritas_diary.odt
```

Данная команда скопирует исходный файл и переименует его. Если в папке, куда производится копирование, уже есть файл с именем **ritas_diary.odt**, он будет перезаписан без предупреждения. Чтобы избежать этого, употребляйте аргумент **-i**: тогда все перезаписи вы будете подтверждать вручную, лично.

Теперь перейдем к потенциально опасной команде **rm**. Она удаляет файлы; однако, в отличие от клавиши Delete графического режима, в текстовом нельзя будет восстановить файл, достав его из Корзины. Нажали клавишу Enter после ввода команды **rm** – и прости-прощай. Чтобы удалить файл, дайте следующую команду:

```
rm /home/bob/vitalfile.ods
```

Можно удалить и сразу несколько файлов:

```
rm /home/bob/vitalfile1.ods vitalfile2.txt vitalfile3.png
```

Резервные копии

Возможности для настройки в Linux весьма обширны, зато легко и угодить в беду, когда система будет работать лишь частично. Предположим, вы хотите вручную отредактировать файл **xorg.conf** (**LXF129**), который расположен в каталоге **/etc/X11** и отвечает за управление работой мыши, клавиатуры и монитора.

Для начала следует создать резервную копию исходного файла **xorg.conf**, командой

```
cp /etc/X11/xorg.conf /etc/X11/xorgbak.conf
```

Затем приступайте к редактированию, будучи уверены, что в случае ошибки вы всегда сможете восстановить систему.

Итак, вы внесли в файл изменение, вышли из системы, перезагрузились, и… И ничего. Из-за какой-то неправильной запятой или точки рабочий стол отказал. Но не все потеряно: ведь у вас есть резервная копия и доступ к командной строке! В подобной ситуации первым делом надо войти в систему от имени суперпользователя-root.

Теперь поменяйте в предыдущей команде источник и цель местами – исходный вариант файла **xorg.conf** восстановится.

```
cp /etc/X11/xorgbak.conf /etc/X11/xorg.conf
```

Вы получите предупреждение насчет перезаписи файла – ну так подтвердите это. Затем перезагрузитесь, и ваши исходные настройки вернутся.

С непривычки командная строка вселяет страх, но предоставляет гибкость и мощь, способную сэкономить вам многие дни работы и гигабайты информации! Освоившись в ней, вы, возможно, захотите установить себе дружелюбную оболочку *Fish* – мы писали про нее в **LXF123**.

Скорая помощь

Если вы хотите увидеть все файлы, в том числе и скрытые, имена которых начинаются с точки (.), пользуйтесь командой 'ls -a'. Здесь опция 'a' означает 'all' – «все».

Просмотр файлов в терминале

Принято говорить, что в системе Linux всё рассматривается как файлы, и многие из них даже человеко-читаемы. Для просмотра содержимого любого файла служит команда **cat**. Например, команда **cat /etc/X11/xorg.conf** покажет текст из файла в терминале. Хорошо, а если текст файла не влезает в окно? Тогда можно воспользоваться командой **less** (точно так же, как и **cat**) и просматривать файл частями. Для перемещения по тексту пользуйтесь клавишами Page Up и Page Down, а для выхода из просмотрщика нажмите Q.

Группы: Рулим

Отличный способ указать, кому и что делать на вашем компьютере.

Linux – многопользовательская операционная система, а значит, в ней можно отделить пространство для себя, своего коллеги или детей, не опасаясь, что кто-то пороется в ваших файлах или испортит ваши любовно созданные обои с умиротворяющим пейзажем. Как администратор вашей домашней (или офисной) системы, вы должны создать другим пользователям возможность комфортной работы, однако не давая им шансов угробить систему. На данном уроке мы рассмотрим создание пользователей и способы контролировать их доступ к компьютеру. Мы используем Ubuntu, но управление пользователями в других дистрибутивах будет очень похожим.

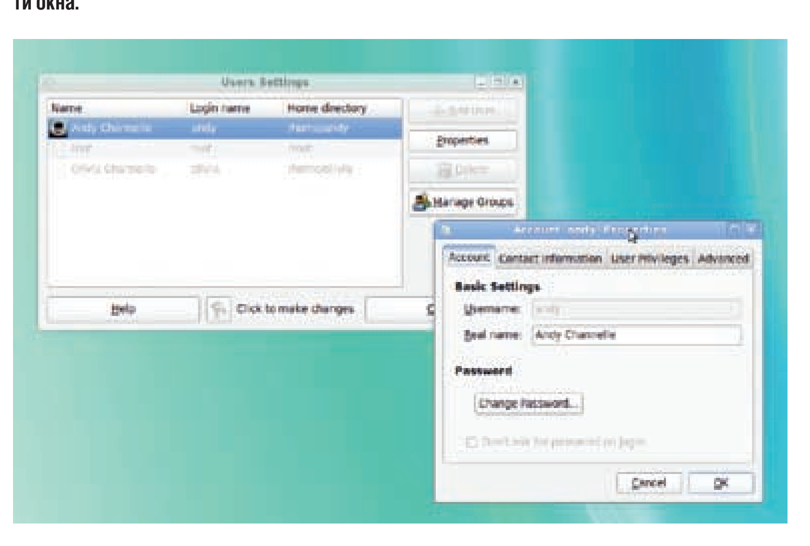
Скорее всего, средства управления пользователями найдутся в меню Система [System] или Администрирование [Administration]. В большинстве дистрибутивов это называется Пользователи и группы [Users and Groups]. Выбор данной опции запустит управляющее приложение, подобное показанному на рисунке внизу.

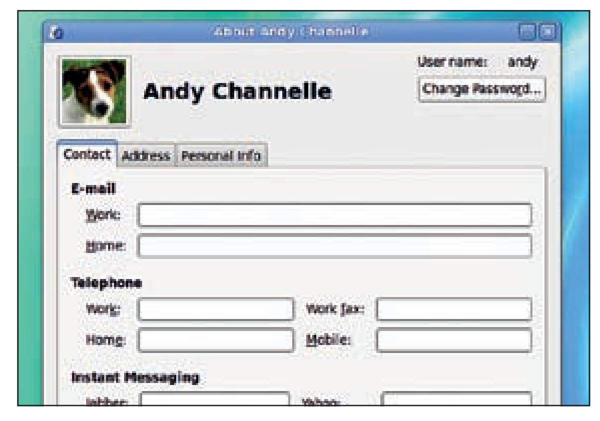
По умолчанию, приложение заблокировано; кнопка Свойства [Properties] активируется, когда вы выберете свое имя пользователя. Итак, выберите его. Сейчас вы вошли в систему как обычный пользователь, и, перелистав вкладки, вы обнаружите, что доступны лишь немногие опции. Вам позволено только сменить свое «реальное имя» (не путайте его с именем пользователя, под которым вы входите в систему) и пароль, добавить кое-какую контактную информацию, и это все.

Для более масштабных изменений, щелкните мышью по значку с изображением ключа в нижней части окна. Система запросит у вас пароль пользователя root (или, если у вас Ubuntu, пароль команды *sudo*, см. **LXF129**). Введя пароль, снова щелкните по кнопке Свойства – вы увидите, что доступных опций стало гораздо больше. Ими мы займемся чуть позже, а пока нажмите кнопку Добавить пользователя [Add User].

Когда появится окно настройки, вы сможете ввести информацию о пользователе на вкладке Учетная запись [Account] – в том числе неизменяемое имя пользователя [username], «реальное имя» [real name] и пароль. Добавив новое имя пользователя, загляните на вкладку Дополнительно [Advanced] и убедитесь, что «в честь него» будет назван и каталог пользователя по умолчанию (он расположен в **/home**).

› **Чтобы внести изменения или добавить новых пользователей, щелкните по значку с ключом в нижней части окна.**





› **Обычные пользователи могут разве что сменить свою картинку на экране входа.**

Проделайте эту процедуру для всех пользователей, которым вы хотите разрешить работать за вашим компьютером. Каждый пользователь получит личное пространство, и изменения, вносимые им в свои рабочий стол, приложения или настройки системы, ограничатся этим рабочим столом.

Все пользователи – включая и вас – могут менять картинки на своем экране входа; можно выбрать ее из системного списка, а можно назначить свою. Для этого перейдите в меню Система > Настройки > Обо мне [System > Preferences > About Me].

Создавая пользователей, вы можете назначить для них набор ограничений. Самый либеральный набор – Администратор [Administrator]: такие пользователи получат те же права, что и вы. Вариант Пользователь [Desktop user] запрещает устанавливать программы и менять что-то вне своего каталога **/home**. Для непривилегированных (‘Unprivileged’) пользователей вы можете создать во вкладке Привилегии [User Privileges] любой набор прав на свое усмотрение.

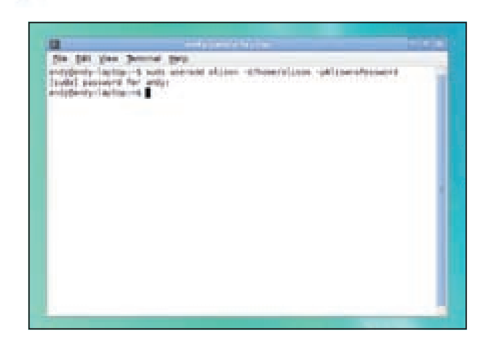
Польза от групп

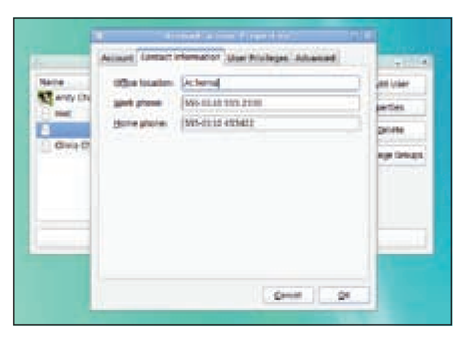
Наличие в системе множества пользователей выгодно тем, что каждый из них получает свое пространство. А вот система объединения пользователей в группы [Groups] уже не столь интуитивна. Зато группы упрощают управление большим количеством пользователей: не нужно настраивать каждого отдельно. Так, группе Adults [Взрослые] можно дать доступ на чтение/запись к неким каталогам, а пользователей из группы Kids [Дети] туда не допускать. Подробнее об этом см. на соседней странице.

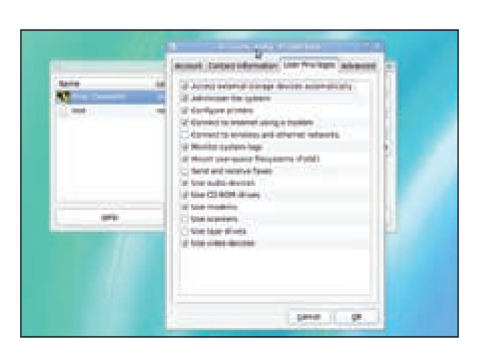
пользователями



Шаг за шагом: Управляем пользователями и группами







1 Добавим пользователя

Команда, создающая нового пользователя – **useradd**. Она требует привилегий root, поэтому перед ее вводом дайте команду **su**, или предварите ее префиксом **sudo**. Чтобы обустроить нового пользователя, введите

```
useradd имя_пользователя -d /home/alison -p
PassWord1.
```

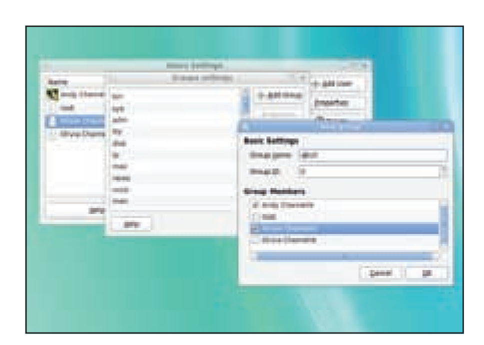
Ключ **-d** создает домашний каталог, а ключ **-p** задает для пользователя пароль. По умолчанию, новый пользователь будет полностью непривилегированным.

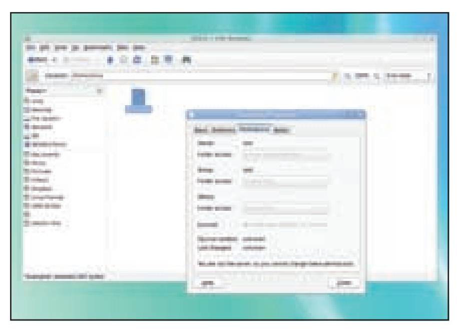
2 Добавим контакты

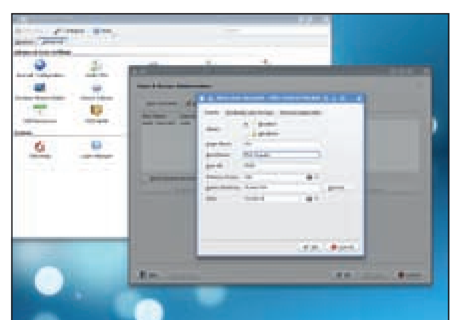
Многие системы Linux используют контактную информацию пользователей в приложениях наподобие *Evolution*, так что перейдите на вкладку Контактная информация [Contact Details] и заполните соответствующие поля. Это – одна из опций, доступных конечным пользователям, и вы можете отдать ее на откуп им самим.

3 Редактируем настройки

На вкладке Привилегии можно ограничить доступ пользователей к устройствам, например, CD-приводу, принтеру или сканеру. Чтобы работать в ней, вы должны войти как root (или аутентифицироваться с помощью пароля **sudo**). Обычные пользователи не могут здесь ничего менять. Для выбора сервиса, выделите его щелчком мыши; повторный щелчок сбросит выделение.







4 Создадим группы

В окне Настройки пользователя [User Settings] выберите опцию Управление группами [Manage Groups], а затем щелкните мышью по кнопке Добавить группу [Add Group]. Появится новое диалоговое окно, где можно дать новой группе имя (допускаются только строчные буквы), после чего можно будет указать членов группы, используя флажки слева от имен пользователей.

5 Удалим пользователя/группу

Если вы насоздавали слишком много пользователей и групп, вы можете решить ликвидировать часть из них. Тогда снова выполните аутентификацию, щелкните по значку с изображением ключа, выберите имя лишнего пользователя и нажмите кнопку Delete. Учтите, что при удалении пользователя его домашняя папка не уничтожается. Все файлы в ней получат нового владельца – пользователя root, и для их удаления понадобится повышение уровня привилегий.

6 Пользователи в KDE

В KDE все эти процедуры точно такие же, только интерфейс выглядит несколько иначе. Выберите в главном меню пункт Компьютер > Параметры системы [Computer > System Settings], затем Дополнительно [Advanced] и Управление пользователями [User Management]. Там присутствуют все опции, которые мы рассмотрели, включая функции для создания и удаления пользователей и групп. Нужно будет опять-таки зарегистрироваться как root или дать команду sudo. LXF

Через месяц Родительский контроль и настройка рабочего стола.

gLabels: Вино как

Любитель винца **Мо Авкати** экономит наличные, варит вкусный напиток и венчает плоды своих трудов несложной свободной программой.





Наш эксперт

Мо Авкати засиделся в информационных менеджерах в госсекторе, и теперь желает, чтобы все общественные организации использовали Linux.

Учитывая последствия кредитного кризиса, остро ощущаемые до сих пор, плюс рост налогов на алкоголь, нечего и удивляться, что все больше народу следует по стопам Тома и Барбары из *The Good Life* и гонит спиртные напитки на дому. Отчего же не попробовать свой способ, получив удовольствие от создания личной марки вина? Более того, ничто более не соответствует духу самодостаточности, чем применение для этого программы с открытым исходным кодом – типа *gLabels*.

gLabels – легковесное приложение рабочего стола Gnome для создания CD/DVD этикеток, визиток, наклеек с адресами и даже штрих-кодов. В комплекте с пакетами вашего дистрибутива у вас, скорее всего, будет версия 2.2.5. Однако версия 2.2.7 появилась на серверах в марте 2010 года и успела попасть к нам на **LXF**DVD.

Как правило, этикетки размещаются на листах А4 и оптимизируются для печати на струйных, лазерных принтерах (или и тех, и других). В шикарных канцелярских магазинах выбор этикеток неплох; но если вы ищете что-то эдакое, пошарьте в Интернете – там немало поставщиков нестандартных вариантов. Чуть-чуть потрудившись, вы найдете массу новых идей, советов и ресурсов для создания этикеток.

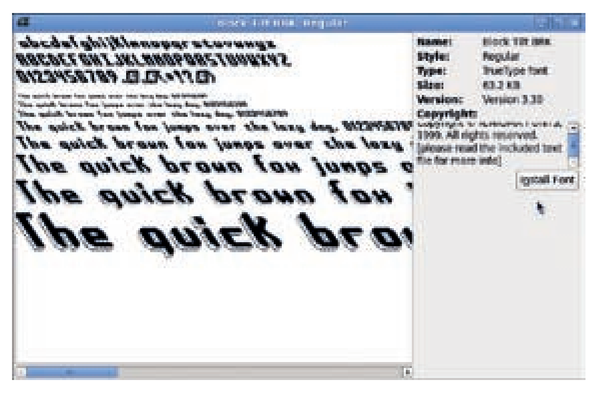
Этикетки для нашего урока взяты из источников Ryman Stationers (www.ryman.co.uk), с использованием его собственного брэнда. Шаблон классифицируется как P8, и он совместим с шаблоном L7165 Avery (www.avery.co.uk). Не обнаружив шаблон в списке *gLabels*, можно создать собственный, с помощью конструктора. Откройте его через меню Файл > Конструктор шаблонов [File > Template Designer]. Конструктор проведет вас через весь процесс. Возможность быстро получить размеры вашего макета существенно упрощает жизнь. На этом этапе можно изменить размеры этикетки, если вам не нравится белая рамка вокруг нее.

Подгонка шрифтов

Если вы хотите делать этикетки, важно иметь хорошую библиотеку шрифтов – они пригодятся и для издательского дела. Хорошие места для поиска шрифтов – www.1001freefonts.com/fonts/afonts2.htm и www.acidfonts.com/index.html. Большинство шрифтов доступны бесплатно для личного использования, ну и, конечно, в менеджере пакетов вашего дистрибутива шрифты тоже имеются.

В большинстве дистрибутивов установить шрифты просто. Один из методов заключается в создании папки шрифтов в вашей домашней директории – назовите ее **.fonts**. Каталоги, имена которых начинаются с **.** (точка) – скрытые, и для просмотра их через стандартный файловый менеджер вы должны отметить галочкой пункт Показывать скрытые файлы [Show Hidden Files] в меню Вид [View]. Теперь остается только скопировать файлы шрифтов в эту папку. Пользователям Fedora повезло – по умолчанию, у них есть установщик шрифтов: в своем браузере нажмите кнопку, чтобы загрузить шрифт, и выберите Открыть с помощью *Archive Manager*. Откройте файл шрифта двойным щелчком мыши, а затем левой кнопкой нажмите на кнопку Установить [Install] справа. Шрифты скопируются в папку, которую вы только что создали (этот метод также создает папку **.fonts** в вашем домашнем каталоге, если ее еще нет).

Теперь давайте изготовим этикетку для вашей домашней бражки.



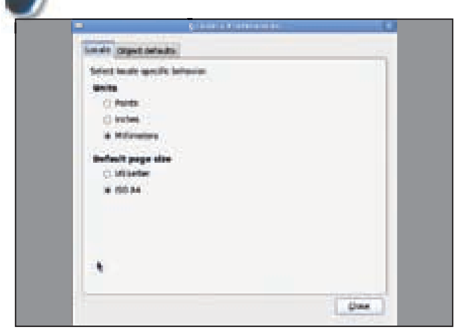
› **Fedora 12 предлагает простой способ установки шрифтов. Вы угадали: достаточно нажать кнопку Установить.**

В кино



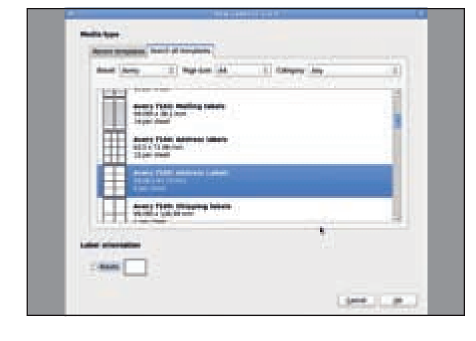


Шаг за шагом: Создаем этикетку для вина



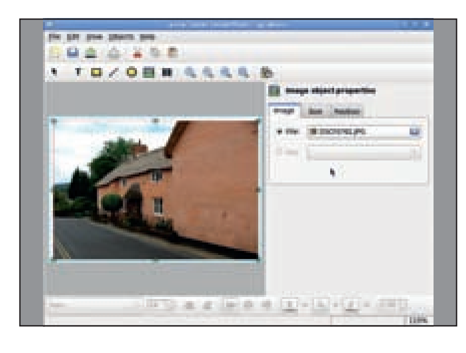
1 Настройки по умолчанию

Запустите *gLabels*, перейдите в Правка > Настройки [Edit > Preferences] и укажите формат листа бумаги и размеры, которые вы хотите использовать. При первом запуске, настройки по умолчанию – формат US Letter, а единицы измерения – дюймы, но вы можете изменить это на ISO A4 и миллиметры, поскольку в большинстве этикеток используются метрические измерения. На этом этапе также можно задать шрифт, размер, цвет и прочие параметры по умолчанию для новых объектов. Неплохо будет освоиться в приложении до того, как вы измените эти настройки.



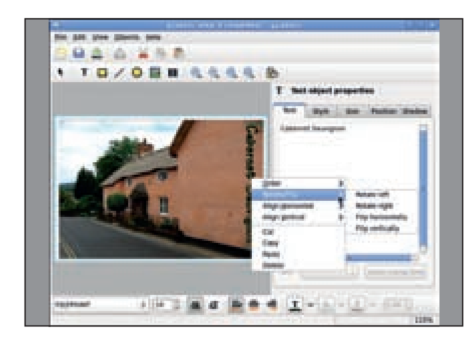
2 Выбор шаблона

Выберите Файл > Создать [File > New] или нажмите на значок Новый [New], и вы увидите экран Тип бумаги [Media Type]. Здесь находятся стандартные шаблоны, а также те, что создали вы. Можно также поискать шаблоны по производителю и типу. При создании дизайна этикетки пригодится кнопка Положение [Orientation]: она позволяет менять ориентацию страницы с книжной на альбомную.



3 Импорт фотографий

Выбрав шаблон, вы увидите пустую этикетку, в качестве основы для вашей. Выберите Объект > Создать > Изображение [Object > Create > Image] или нажмите на значок на панели и выберите Создать объект-изображение [Create Image object], перейдите к вашей библиотеке фотографий и укажите снимок для этикетки. Поддерживаются основные графические форматы, включая SVG, ICO, PNG и TIFF. Размеры изображения изменяются при помощи рукояток по его углам, что исключает искажение пропорций.



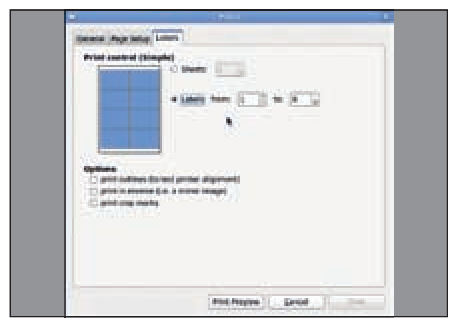
4 Добавление текста

Укажите название вашего вина на этикетке, выбрав текстовый объект на панели инструментов. На данном этапе можно задать шрифт, кегль и цвет текста. Для точного положения объекта, выделите его и используйте клавиши стрелок на клавиатуре. Теперь правой кнопкой мыши нажмите на текстовом объекте и выберите пункт Повернуть/отразить > Повернуть вправо [Rotate/Flip > Rotate Right]. К сожалению, возможности отмены нет – имейте это в виду. Заведите привычку щелкать на сером фоне вокруг рабочей области, отменяя выделение, чтобы случайно не удалить или переместить объект.



5 Сделайте профессионально

Добавьте два текстовых объекта, один для даты и один – по содержанию алкоголя (и пингвина не забудьте!). Приведите шрифт и цвет текста в гармонию с вашим дизайном. Можете добавить глубины, отметив галочкой Включить тень [Enable shadow]; прозрачность и цвет эффекта теней тоже настраиваются. Нанесите завершающий штрих на этикетку, вставив штрих-код. Для этого нажмите на значок штрих-кода на панели инструментов, затем щелкните по рабочей области, чтобы создать штрих-код объекта. Выберите вкладку Style [Стиль], затем Code 128 [Код 128] и вкладку Data [Данные], после чего введите название вашего виноградника напротив Литерал [Literal data type] в текстовом поле.



6 Время печатать

Наконец пришло время печатать этикетки. Используйте опцию печати набросков, чтобы проверить выравнивание, особенно если вы применяли самодельный шаблон. Параметры печати в gLabels позволяют указать точку начала печати, что полезно при печати нескольких этикеток из всего листа, или можно использовать весь лист для печати более одного типа этикетки: т.е. сначала напечатать один, потом второй и т.д. Сохранение листа нетронутым снизит риск замятия бумаги в принтере. Теперь наслаждайтесь выпивкой стильно! LXF

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

apt-get: Хозяин

Мечтали о едином средстве управления пакетами и «околопакетным» хозяйством? **Энди Хадсон** представляет дивный инструмент по теме.





Наш эксперт

Энди Хадсон прирожденный хакер. Написал несколько книжек о Linux. Если мы не можем в чем-то разобраться, то идем к нему.

В зависимости от рода своей деятельности в мире Linux, кто-то предпочитает *portage*, кто-то – *Yum*, кто-то просто собирает приложения из исходников вручную. Однако в растущей массе дистрибутивов на базе Debian есть только один правильный способ управления пакетами: *apt-get*. Нетрудно понять, почему племя дебианцев так почитает этот инструмент: *apt-get* был их оплотом много лет. Программа быстрая и удивительно простая в работе – при бесподобной функциональности.

› **По окончании работы apt-get update в окне вашего терминала должен появиться перечень вроде этого.**

Большинство пользователей не вникает в тонкую механику системы, поэтому на нашем уроке мы решили рассказать вам, как извлечь из этого мощного инструмента максимум и во мгновение ока превратиться из новичка в профессионала.

Предполагается, что вы пока не работали с *apt-get*: люди чаще используют *Synaptic*, стандартную графическую программу для Debian и Ubuntu. Итак, перейдите в командную строку и готовьтесь к бою. Поскольку *apt-get* проводит изменения системного масштаба, вам понадобится стать суперпользователем-root или получить его права, иначе сделать что-либо не удастся. На данном уроке мы будем считать, что вы вошли от имени root или (для Ubuntu) предваряете каждую команду префиксом **sudo**, чтобы получить необходимые привилегии. Итак, сначала наберите

```
apt-get update
```

Команда предписывает *apt-get* обновить информацию об имеющихся пакетах с помощью сведений, взятых из файла **sources.list**. Это необходимо: прежде чем двигаться дальше, следует получить наиболее свежую информацию о пакетах, доступных на текущее время. По окончании работы *apt-get* можно вводить команду

```
apt-get upgrade
```

Будут загружены самые свежие версии установленных приложений, вместе с зависимостями. Сопутствующие действия выполняются автоматически. Кроме того, на данном этапе нужно усвоить еще две базовых функции *apt-get*. Вот они:

```
apt-get install нечто
```

Эта команда добавляет приложение, а следующая – удаляет:

```
apt-get remove нечто
```

Теперь, зная базовые команды, рассмотрим возможности *apt-get* подробнее.

Поддерживаем чистоту

Начнем с того, что, хотя *apt-get* очень эффективный менеджер пакетов, со временем он образует немало «хлама», который не всегда удаляется системой *Apt* автоматически. Этот «хлам» состоит главным образом из пакетов, которые прошли процесс обработки и установки, но остались лежать в файловой системе.

Если система «молодая», то вы этой проблемы и не заметите (особенно если часто меняете дистрибутивы). Однако если у вас установлен Debian или одна из версий Ubuntu LTS, то регулярная уборка будет полезной. Просто наберите команду

```
apt-get clean
```

Apt уберет ненужные пакеты и освободит вам изрядный кус дискового пространства, особенно если вы давно не занимались очисткой или провели крупную серию обновлений. Единственный недостаток данного способа – если понадобится переустановка пакета, *apt-get* придется загрузить его заново. Чтобы этого избежать, когда о дисковом пространстве беспокоиться незачем, наберите такую команду:

пакетов

Чушь... коровья

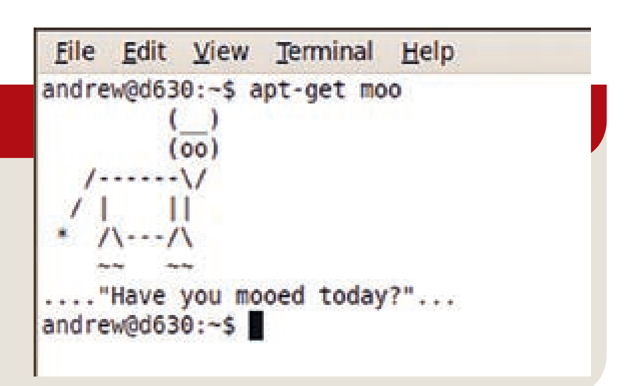
Если вы введете команду **apt-get** «как есть», то получите обычный перечень доступных параметров, часто в сопровождении шифрованной фразы This apt has Super Cow Powers («У этого apt суперкоровья сила»). Узреть свет поможет команда

```
apt-get moo
```

Вы увидите ASCII-изображение коровы и вопрос, который непременно вызовет у вас улыбку. У *Aptitude*, компаньона *apt-get* более высокого уровня, есть подобная функция – попробуйте набрать

```
aptitude moo && aptitude moo -v
```

и полюбуйтесь результатом. Теперь добавляйте лишнее **v** в конец второй команды – вы оцените юмор.




```
apt-get autoclean
```

Будут удалены только лишние пакеты. Например, если вы недавно несколько раз обновляли какое-либо приложение, **autoclean** избавит вас от всех версий, кроме самой свежей (она может пригодиться для переустановки). Такой способ очистки стоит применять при частой переустановке приложений, даже если у вас быстрое интернет-подключение.

Для сверхчистюль – еще одна тонкость: использование переключателя **purge** вместо **remove**. Такая команда удаляет не только лишние пакеты, но и все конфигурационные файлы, использовавшиеся при установке.

Заключительная строка в очистительном арсенале *apt-get* –

```
apt-get autoremove
```

Команда **autoremove** удаляет пакеты, которые были загружены в качестве зависимостей, но больше не нужны.

Разобравшись с очисткой, поинтересуемся, не пригодится ли *Apt* при компиляции из исходных кодов, «а-ля Gentoo». Оказывается, с помощью *apt-get* компилировать не только можно, но даже предпочтительно: программа освободит вас от лишней мороки. Начнем с получения исходного кода – наберите

```
apt-get source нечто
```

Будут доставлены файлы исходных текстов, включая **.dsc**, который используется *dpkg-source* для пояснения *apt-get*, куда следует распаковать архив. Как правило, исходный код помещается в подкаталог текущей рабочей директории, а его название соответствует имени пакета исходников. Теперь все готово для компиляции обычным манером.

Прикладные фокусы

Если вас не прельщает перспектива ручной компиляции из исходных кодов, то у *apt-get* и тут есть козырь в рукаве: можно предписать программе скомпилировать пакет автоматически, сразу по окончании загрузки.

Приложений-отшельников, без зависимостей, не существует, и перед компиляцией нужно убедиться, что последние установлены. Здесь-то *apt-get* и приносит реальную пользу. Зная название пакета, который вы собираетесь компилировать, достаточно набрать такую команду:

```
apt-get build-dep нечто
```

и все зависимости для названного пакета будут установлены.

Apt-get опросит репозитории, найдет необходимые пакеты и установит их – хотя основной пока не тронет. По завершении этого процесса, наберите команду

```
apt-get source нечто -b
```

Исходный код будет доставлен и скомпилирован. Но не установлен: *apt-get* лишь формирует пакет **deb** в текущей рабочей директории. Устанавливать его придется самостоятельно – для этого воспользуйтесь командой **dpkg**:

```
dpkg -i название_пакета.deb
```

Поскольку о зависимостях мы с вами уже позаботились, пакет будет установлен, и – вуаля, пользуйтесь на здоровье! LXF

› Прежде чем приступать к компиляции пакета из исходных текстов, позаботьтесь о наличии зависимостей.

```
andrew@d630: ~
File Edit View Terminal Help
gpgv: Can't check signature: public key not found
dpkg-source: warning: failed to verify signature on ./cheese_2.28.1-0ubuntu1.dsc
dpkg-source: info: extracting cheese in cheese-2.28.1
dpkg-source: info: unpacking cheese_2.28.1.orig.tar.gz
dpkg-source: info: applying cheese_2.28.1-0ubuntu1.diff.gz
dpkg-buildpackage: set CFLAGS to default value: -g -O2
dpkg-buildpackage: set CPPFLAGS to default value:
dpkg-buildpackage: set LDFLAGS to default value: -Wl,-Bsymbolic-functions
dpkg-buildpackage: set FFLAGS to default value: -g -O2
dpkg-buildpackage: set CXXFLAGS to default value: -g -O2
dpkg-buildpackage: source package cheese
dpkg-buildpackage: source version 2.28.1-0ubuntu1
dpkg-buildpackage: source changed by Sebastien Bacher <seb128@ubuntu.com>
dpkg-buildpackage: host architecture i386
dpkg-checkbuilddeps: Unmet build dependencies: cdbs (>= 0.4.41) gnome-pkg-tools
(>= 0.10) librsvg2-dev (>= 2.18.0) libgnome-desktop-dev (>= 2.25.1) libgconf2-de
v (>= 2.16.0) libgstreamer0.10-dev (>= 0.10.20) libgstreamer-plugins-base0.10-de
v libebook1.2-dev (>= 1.12.0) libdbus-1-dev (>= 1.0) libhal-dev (>= 0.5.9) libdb
us-glib-1-dev (>= 0.7) docbook-utils
dpkg-buildpackage: warning: Build dependencies/conflicts unsatisfied; aborting.
dpkg-buildpackage: warning: (Use -d flag to override.)
Build command 'cd cheese-2.28.1 && dpkg-buildpackage -b -uc' failed.
E: Child process failed
andrew@d630:~$
```

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

R: Ускоряем на

Завершая мини-курс под кодовым названием «R на примере», **Сергей Петров** и **Евгений Балдин** рассмотрят высокопроизводительные вычисления.



Наш эксперт

Серей Петров
Доцент Гродненского государственного университета им. Я. Купалы, занимается построением и идентификацией моделей системы кровообращения, применением методов многомерного анализа данных. Администрирует Linux с 1994 года. Крайне сожалеет, что начал использовать R только в 2001 году.



Наш эксперт

Евгений Балдин
Физик, преподаватель, научный сотрудник ИЯФ им. Будкера, профессионально занимается обработкой экспериментальных данных в области физики высоких энергий. Давний сторонник свободного ПО, очарованный мощью R.

Компьютеры ускоряются, диски становятся просторнее, но темпы роста объёма данных и усложнение их обработки всё равно вынуждают ускорять вычисления.

По традиции, сложные вычисления выполняют кластеры. Многие наверняка слышали про список TOP500, но речь пойдет не о нём: кластерные технологии потихоньку проникают и на обычные компьютеры, можно сказать, в дома. Почти у каждого процессора сейчас несколько ядер. Отчего же этим не воспользоваться? Но сперва прикинем, как оценить выигрыш от подобных приёмов.

Анализ эффективности

Измерить скорость вычислений в R и, соответственно, оценить эффективность написанного кода можно несколькими способами:

- Применить system.time() для простых измерений.
- Профилировать выполнение кода с помощью Rprof().
- Профилировать потребление памяти с помощью Rprofmem().

Для визуализации накопленных с помощью Rprof() данных можно применять пакеты **profr** и **proftools**.

Приложим средства профилирования к простой задаче. Пусть в ходе моделирования многократно вычисляются параметры линейной регрессии, через функцию lm(). Её недостаток – много лишних действий: она просто слишком универсальна. Для простой линейной регрессии достаточно уточнённого вызова lm.fit().

Насколько же эффективнее прямой вызов? Попробуем обработать набор макроэкономических показателей longley с помощью обоих функций. Построим зависимость между числом работающих и остальными показателями. Для оценки проделаем по 1000 вычислений за раз.

```
# Загружаем данные
> data(longley)
# Записываем профиль в файл lm.out
> Rprof(“lm.out”)
# Выполняем lm() 1000 раз
> invisible(replicate(1000,lm(Employed ~ .-1, data=longley)))
# Отключаем профилирование
> Rprof(NULL)
# Готовим данные для lm.fit()
> longleydm <- data.matrix(data.frame(longley))
# Записываем профиль в файл lm.fit.out
> Rprof(“lm.fit.out”)
# Выполняем lm.fit() 1000 раз
> invisible(replicate(1000,lm.fit(longleydm[,-7],longleydm[,7])))
# Отключаем профилирование
> Rprof(NULL)
```

Записанные в файлах профилирования данные можно проанализировать с помощью встроенной команды summaryRprof(). Например, для того, чтобы выяснить время работы программы, достаточно обратиться к переменной sampling.time:

```
> summaryRprof(“lm.out”)$sampling.time
[1] 6.42
> summaryRprof(“lm.fit.out”)$sampling.time
[1] 0.44
```

Данные профилирования можно отобразить и графически (см. рисунок), с помощью пакета **profr**:

```
# Устанавливаем пакет (если нужно)
> install.packages(“profr”)
> library(“profr”)
> plot(parse_rprof(“lm.out”),main=”Profile of lm()”)
> plot(parse_rprof(“lm.fit.out”),main=”Profile of lm.fit()”)
```

Из рисунков видно, сколько времени проводит программа в каждой из функций. Неудивительно, что lm.fit() работает в четырнадцать (!) раз быстрее.

Для представления зависимостей вызовов в виде графа можно воспользоваться пакетом **proftools**. Для этого необходимо установить системный пакет **graphviz-dev**:

```
=> aptitude install graphviz-dev
```

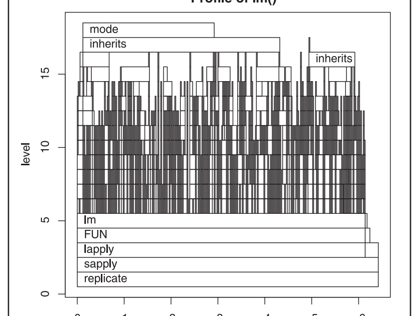
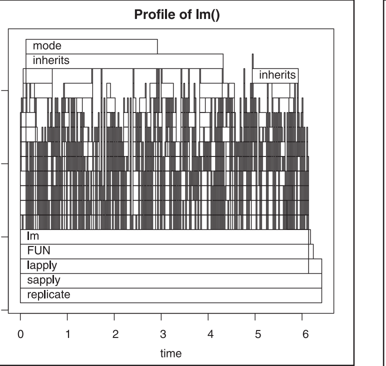
а также пакет **Rgraphviz** и сам **proftools**:

```
> source(“http://bioconductor.org/biocLite.R”)
> biocLite(“Rgraphviz”)
> install.packages(“proftools”)}
```

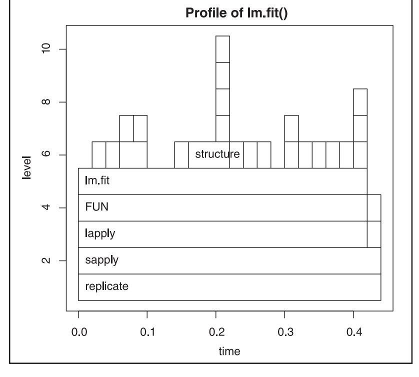
Для отображения графа нужно выполнить такие команды:

```
> library(“Rgraphviz”)
> library(“proftools”)
> lmfitprod <- readProfileData(“lm.fit.out”)
> plotProfileCallGraph(lmfitprod)
```

Для отладки потребления памяти нужно использовать специально модифицированную версию R, которая собрана с опцией **--enable-memory-profiling**. При этом для анализа применяется команда Rprofmem, и все происходит аналогично использованию Rprof.

Результаты профилирования функции lm()...



...и lm.fit(). Как говорится, почувствуйте разницу!

Месяц назад Мы создали для приложений R графический интерфейс.

примере

Перерасход дисковой памяти отследит функция tracemem: она срабатывает при каждом копировании какого-либо объекта.

Ключ к ускорению

Достичь максимальных скоростей поможет знание встроенных функций R. Обычно такие функции создаются, когда выполняемые ими задачи начинают частить в обработке данных. Подобный подход существенно сокращает усилия по вводу кода программы, а также ускоряет выполнение кода, за счёт реализованных во встроенных функциях оптимальных алгоритмов.

Как правило, встроенные функции реализованы на более низкоуровневом языке, предоставляющем углубленный доступ к возможностям оборудования. Поясним это на примере: напишем программу сложения чисел от 1 до указанного, в обычном цикле, и сравним с программой, использующей встроенные функции R:

```
> mysum <- function(N) { a <- 0;
+   for (i in 1:N) a <- a + i;
+   return(a) }
> system.time(mysum(10000000))
   user  system elapsed
  7.456   0.024   7.482

> system.time(sum(as.numeric(seq(1,10000000))))
   user  system elapsed
  0.052   0.060   0.112
```

Разница очевидна: второй вариант подсчёта суммы работает более чем в 70 раз быстрее! То, что сейчас было проделано, носит название «векторизация». На такой способ расчётов трудно переключиться после императивного стиля программирования, но именно векторизация обуславливает необычайно высокую продуктивность работы в R. Заметьте: не только уменьшилось время выполнения программы, но и сократился её размер!

Возьмем простую тестовую задачу: «Найти распределение детерминанта матрицы 2×2, в которую занесены независимо и случайно изменяющиеся значения из диапазона 0, 1, 2, … , 9». Она эквивалентна задаче о нахождении всех сочетаний ab-cd, где a, b, c, d – это цифры.

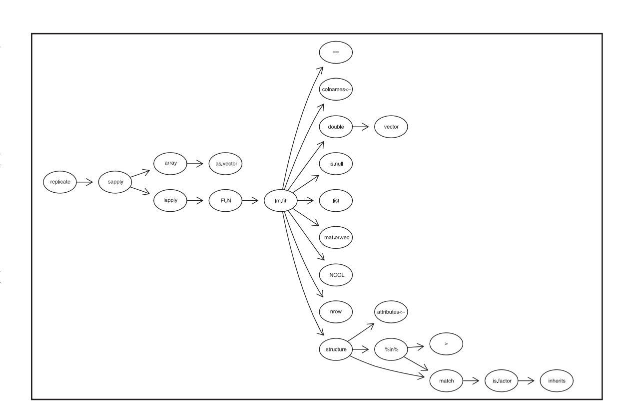
Наивное императивное решение «с циклами» выглядит привычно, и благодаря синтаксису R достаточно «компактно»:

```
> dd.for.0 <- function()
+ {
+   val <- NULL
+   for (a in 0:9) for (b in 0:9) for (d in 0:9) for (e in 0:9)
+     val <- c(val, a*b - d*e)
+   table(val)
+ }
> system.time(dd.for.0())
   user  system elapsed
  0.196   0.000   0.195
```

От запуска к запуску время слегка изменяется, так что распределение отличается от нормального, и лучше всего находить медиану нескольких попыток:

```
median(replicate(20, system.time(dd.for.0())["elapsed"]))
[1] 0.177
```

Попробуем добиться от этого наивного варианта большего: например, выделим память под расчёты сразу в начале программы.



› **Граф вызовов, полученный с помощью пакета proftools.**

```
> dd.for.1 <- function()
+ {
+   val <- double(10000)    # преаллоцируем val
+   nval <- 0
+   for (a in 0:9) for (b in 0:9) for (d in 0:9) for (e in 0:9)
+     val[nval <- nval + 1] <- a*b - d*e
+   table(val)
+ }

> median(replicate(20, system.time(dd.for.1())["elapsed"]))
[1] 0.059
```

Эффект от проделанных изменений очевиден: код ускорился более чем в три раза. Поскольку наши данные – целые числа, давайте посмотрим, что даст нам использование встроенной функции tabulate():

```
> dd.for.3 <- function()
+ {
+   val <- double(10000)
+   nval <- 0
+   for (a in 0:9) for (b in 0:9) for (d in 0:9) for (e in 0:9)
+     val[nval <- nval + 1] <- a*b - d*e
+   tabulate(val)
+ }
> median(replicate(20, system.time(dd.for.3())["elapsed"]))
[1] 0.057
```

Время выполнения сократилось еще чуть-чуть, но это всё полумеры. Давайте сделаем решительный шаг и вспомним, что прародителем R был и язык APL. Запишем решение задачи как операцию над массивами:

```
> dd.fast <- function()
+ {
+   val <- outer(0:9, 0:9, "*")
+   val <- outer(val, val, "-")
+   tabulate(val)
+ }

> median(replicate(20, system.time(dd.fast())["elapsed"]))
[1] 0.001
```

Лучшее из наших решений, использующих циклы, обойдено более чем в 50 раз, а «традиционное» – почти в 200!

Что же делать, если избавиться от цикла нельзя? Один из вариантов – использовать среду R, в которую встроена возможность »

компиляции кода. Сборка R с функцией JIT (Just-in-Time Compilation, компиляция во время выполнения) позволяет рассчитывать на ускорение кода, содержащего циклы, раза в полтора.

Операции с матрицами являются в R такими производительными, потому что они опираются на процедуры библиотеки *BLAS* (Basic Linear Algebra Subprogram). R может быть скомпилирована с различными вариантами реализации *BLAS*: это и свободная библиотека *Atlas* (пакет *atlas3-base*), и платная *Goto*, и библиотеки от двух основных производителей процессоров: Intel (Intel Math Kernel Library, http://software.intel.com/ru-ru/intel-mkl/) и AMD (AMD Core Math Library, http://www.amd.com/acml). Они не только имеют более производительный код, но и автоматически задействуют в вычислениях все имеющиеся ядра процессора персонального компьютера. Дополнительный прирост производительности можно получить, настроив *Atlas* под конкретно используемый в расчётах процессор.

Помимо библиотеки BLAS, можно «попытать счастья» с экспериментальным пакетом *pnmath0* (http://www.stat.uiowa.edu/~luke/R/experimental/) от Люка Тьерни [Luke Tierney]. Он распараллеливает выполнение векторных функций R, используя потоки Pthreads. Эта возможность пока не встроена в R – её придётся добавить самостоятельно. Учтите, что параллельные вычисления активируются только при достаточной длине векторов аргументов.

Если на компьютере установлена видеокарта, которая поддерживает вычисления на своем GPU (пока сюда входят только NVIDIA CUDA и CUBLAS), то при установке пакета *gputools* появляется возможность выполнять иерархический кластерный анализ, классификацию с обучением (по алгоритму SVM) и расчёт коэффициентов корреляции с очень высокой скоростью.

Несмотря на использование высокопроизводительных векторных операций и компиляции в режиме Just-in-Time, бывают моменты, когда на счету каждый такт процессора. В этом случае есть два способа оперативно встроить в расчет, выполненный в среде R, низкоуровневый код императивного языка программирования:

» **inline** — для простой вставки небольшого фрагмента кода;
» **Rcpp** – для облегчённого процесса интеграции сложного кода на C++.

Пакет inline предоставляет функцию cfunction(), умеющую автоматически встраивать код, написанный на Fortran, C, C++. Для выполнения следующего простого примера на Fortran, естественно необходимо установить и загрузить сам inline:

```
# Не забудьте про отступы! Fortran есть Fortran.
> code <- “
+      do i = 1, n(1)
+       x(i) = x(i)**3
+      end do”
> cubefn <- cfunction(signature(n=”integer”, x=”numeric”),
+                  code, convention=”.Fortran”)
> x <- as.numeric (1:10)
> n <- as.integer(10)
> cubefn(n,x)$x
[1]    1    8   27   64  125  216  343  512  729 1000
```

В параллель

Среда R, работающая в 64-битном окружении, практически не имеет ограничений на объём обрабатываемых данных. Современным ответом в области вычисления с гигантскими массивами данных является пакет *iterators* от REvolution Computing. С его возможностью поэлементно обработать структуру, не вмещающуюся в память, крайне удачно сочетается второй пакет, *foreach*. Он вводит возможность циклически обработать созданный итератор и вернуть суммарный результат. Отсутствие побочных эффектов позволяет выполнить оптимизируемую операцию параллельно.

Отсутствие побочных эффектов – именно то, что позволяет не заботиться, где выполняется та или иная часть кода. Компьютеры не только стали мощнее и у них теперь больше ядер. Компьютеров, прежде всего, стало *много* больше. Под рукой практически у каждого из нас имеются 5–10 машин, и их многочисленные процессоры, если приглядеться внимательно, не загружены и на 10%. Вся эта мощь доступна пользователю при работе с R.

Кластерные вычисления настолько естественны для векторных операций R, что существует несколько способов их реализации в этой среде:

» **Rmpi** – реализация Message Passing Interface (MPI), который является стандартом в области паралельных вычислений;
» **NWS** – написанная на Python альтернативная реализация MPI;
» **snow** – высокоуровневая надстройка над MPI, PVM, NWS и сокетами (sockets);
» **papply** – параллелизация функции apply через MPI;
» **multicore** – параллельные вычисления на многоядерных машинах.

Поскольку наша цель – это быстро и «глобально» для нашей же пользы задействовать мощь окружающих нас и фактически простаивающих компьютеров, сразу же воспользуемся высокоуровневым средством, а именно, пакетом *snow*. А чтобы рассказ не был пустым теоретизированием, попробуем решить реальную задачу, имеющую практическую ценность.

» Задача

Число телефонных пар, проходящих рядом и передающих сигнал без помех, ограничено. Безусловно, влияет и длина кабеля, и его ёмкость, и диаметр жил. Однако в номограммах оценки характеристик телефонной пары не предусмотрено свыше 25 ADSL-пар в одном кабеле.

Нормативные документы российских провайдеров ADSL определяют емкость одиночного кабеля в 18 %. Скорее всего, для кабелей небольшой ёмкости всё же верен предел на 18 жил, занятых одновременно работающими ADSL-модемами, в одном кабеле. А для кабелей ёмкостью более 100 пар верен процентный лимит.

Пусть в городе Н-ске ёмкость ADSL-сети Н-ск-телеком превысила 10 000 абонентов. Попробуем оценить, какие проблемы встретит сеть при своем развитии.

» Модель

Город Н-ск насчитывает 300 000 населения, в средней семье 3 человека, что определяет следующие исходные условия:

```
# количество абонентов услуги
> n.abonentov <- 10000
# количество домов
> n.domov <- 1000
# количество квартир в доме
> n.kvartir <- 100
# критическое число абонентов для ADSL в одном доме
> n.kritic <- 18
```

Мы получаем модель города в виде вектора, каждый элемент которого – квартира, помеченная номером дома:

```
> gorod <- rep(1:n.domov, each = n.kvartir)
```

Поскольку вычисления ресурсоёмкие, сразу загрузим библиотеку для параллелизации основных функций пакета (предполагается, что эта экспериментальная библиотека уже установлена):

```
> library(“pnmath0”)
```

Делаем выборку случайных n.abonentov в векторе gorod.

```
> vyborka <- as.factor(gorod)[sample(1:length(gorod),
+            n.abonentov, replace= FALSE)]
```

Подсчитываем, сколько абонентов попало в нее в каждом доме, и строим гистограмму.

```
> hist(as.numeric(tapply(rep(1,n.abonentov), vyborka, sum)))
```

Вычислим, сколько домов испытывают трудности:

```
> length(as.numeric(tapply(rep(1,n.abonentov),
+        vyborka, sum))[as.numeric(tapply(rep(1,n.abonentov),
```

```
+                         vyborka, sum))>n.kritic])
[1] 4
```

а также оценим число «проблемных» абонентов:

```
> sum(as.numeric(tapply(rep(1,n.abonentov),
+         vyborka, sum))[as.numeric(tapply(rep(1,n.abonentov),
+                   vyborka, sum))>n.kritic])
[1] 81
```

Это только одна реализация. Построим бутстреп-процедуру (**LXF128**) и оценим долю домов, где будет свыше n.kritic абонентов. Тут нужно не менее 10 000 вычислительных экспериментов:

```
> nn <- 10000
> bstr.dom <- numeric(nn)
> bstr.abonent <- numeric(nn)
> for (n in 1:nn) {
+   vyborka <- as.factor(gorod)[sample(1:length(gorod),
              n.abonentov, replace= FALSE)]
+   rr <- as.numeric(tapply(rep(1,n.abonentov),vyborka , sum))
+   bstr.abonent[n] <- sum(rr[rr>n.kritic])
+   bstr.dom[n] <- length(rr[rr>n.kritic])
+ }
# Проблемные абоненты
> hist(bstr.abonent)
# Проблемные дома
> hist(bstr.dom)
```

Проведём эксперимент с подсчётом: сколько процентов абонентов и какое количество домов окажется с плохим качеством услуги при росте абонентской базы от 10 000 до 20 000 абонентов.

Оформим функцию:

```
> my.boot.adsl <- function (n.abonentov) {
+  nn <- 10000
+  bstr.dom <- numeric(nn)
+  bstr.abonent <- numeric(nn)
+  for (n in 1:nn) {
+   vyborka <- as.factor(gorod)[sample(1:length(gorod),
+                n.abonentov, replace= FALSE)]
+   rr <- as.numeric(tapply(rep(1,n.abonentov),vyborka , sum))
+   bstr.abonent[n] <- sum(rr[rr>n.kritic])
+   bstr.dom[n] <- length(rr[rr>n.kritic])
+  }
+  return(abonent=bstr.abonent, dom=bstr.dom)
+}
```

Рассчитаем оценку числа проблемных абонентов в диапазоне размера абонентской базы от 10 000 до 20 000:

```
> xx <- c(NA)
> for (n in 1:10) {
+  xx[n] <- my.boot.adsl(10000+(n*1000))
+}
```

Отнормируем на размер абонентской базы:

```
> xx.norm <- xx
> xx.norm[[1]] <- xx[[1]]/11000
> for (n in 2:10) {
>  xx.norm[[n]] <- xx[[n]]/(10000+(n*1000))
> }
```

Наконец, отобразим в виде боксплота:

```
> boxplot(xx.norm,names=seq(11000,20000,by = 1000))
```

Данный модельный город предполагал наличие только 100-квартирных домов. Повторим вычисления на данных о реальном количестве квартир в городе Н-ске.

Реальность

Ввиду ресурсоёмкости вычислений напишем параллельную версию программы. Для этого загрузим библиотеку *snow*:

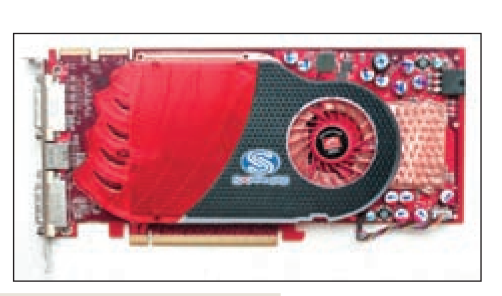
```
> library(snow)
```

Создадим кластер из двух узлов:

```
> cl <- makeCluster(c("loc
alhost","localhost"), type =
"SOCK")
```

Критическое число абонентов для ADSL в одном доме –

```
> n.kritic <- 18
> clusterExport(cl, "n.kritic")
```



Современные видеокарты демонстрируют производительность, сравнимую с мощнейшими суперкомпьютерами десятилетней давности.

Получаем модель города в виде вектора, каждый элемент которого – квартира, помеченная номером дома. Обработаем реальные дома города Гродно (Беларусь). Загрузим список максимальных номеров квартир в доме:

```
> Nsk <- na.omit(read.table("data_Nsk.txt"))
> Nsk.sort <- sort(t(Nsk))
```

Файл data_Nsk.txt – просто колонка чисел, которую можно скачать, например, отсюда: http://www.inp.nsk.su/~baldin/data_Nsk.txt.

Создадим модель города:

```
> gorod <- unlist(mapply(rep,
+      1:length(Nsk.sort),Nsk.sort))
> clusterExport(cl, "gorod")
```

Отведём место под результаты оценки числа проблемных домов и абонентов, для уменьшения накладных расходов при выделении памяти.

```
> nn <- 10000
> bstr.dom <- numeric(nn)
> clusterExport(cl,"bstr.dom")
> bstr.abonent <- numeric(nn)
> clusterExport(cl,"bstr.abonent")
```

Функция расчёта числа проблемных абонентов при случайном распределении выглядит так:

```
> my.boot.func <- function (n, n.abonentov) {
+ vyborka <- as.factor(gorod)[sample(1:length(gorod),
+                n.abonentov, replace= FALSE)]
+ rr <- as.numeric(tapply(rep(1,n.abonentov),vyborka , sum))
+ bstr.abonent[n] <- sum(na.omit(rr[rr>n.kritic]))
+}
```

Рассчитаем оценку числа проблемных абонентов в диапазоне абонентской базы от 3000 до 32 000 человек

```
> xx <- cbind(sapply(1000+((1:10)*3000),
+               function (n.abonentov) parSapply(cl,
+                  c(1:nn), my.boot.func, n.abonentov )))
```

Вновь отнормируем на размер абонентской базы:

```
> xx.norm <- xx
> xx.norm[,1] <- xx[,1]/4000
> for (n in 2:10) {
+ xx.norm[,n] <- xx[,n]/(1000+(n*3000))
+ }
```

И, наконец, отобразим в виде боксплота, после чего остановим кластер.

```
> boxplot(xx.norm, names=seq(4000,31000,by = 3000))
> stopCluster(cl)
```

А в чём выигрыш?

По приведённым выше вычислениям тестирование выполнялось на процессоре семейства Intel Core Duo, модель T2050 с частотой 1,60 ГГц. При использовании распараллеливания время вычисления составляло 6470 секунд, а без оного – 12754 секунд. Иными словами, два ядра примерно в два раза лучше, чем одно.

Что и Требовалось Доказать. LXF

Через месяц Мы завершим практикум по R яркими примерами.

OOo Calc: Агенты

Создадим имитационную агентную модель в OOo Calc вместе с **Алексеем Джежорой**.

LINUX FORMAT на DVD
» Таблица Calc с нашей моделью



Наш эксперт

Алексей Джежора периодически заставляет приложения работать на пределе возможностей, от чего и страдает.

Адепты определяют моделирование как основной метод познания функционирования систем. И, коль скоро система есть форма организации всего сущего, то без особого преувеличения можно сказать, что моделирование – это главный способ отражения действительности.

Теория выделяет различные виды моделирования: от естественно-языкового описания до имитационной компьютерной реализации, при которой логико-математический алгоритм функционирования системы исследуется посредством компьютерной программы-имитатора. В свою очередь, компьютерное имитационное моделирование может выполняться различными методами, среди которых присутствует метод агентного моделирования. Его особенностью является то, что имитационная модель представляет собой совокупность взаимодействующих отдельных программных или программно-аппаратных сущностей – агентов, обладающих индивидуальными целями, правилами внутреннего и внешнего «распорядка». По такому принципу удобно моделировать как поведение стада парнокопытных, так и работу производственного предприятия, а также много чего/кого другого.

Существует свободное ПО, адаптированное для задач агентного моделирования: например, пакет *Swarm* (www.swarm.org), используемый для моделирования социально-биологических и социально-экономических систем; *breave* (http://www.spiderland.org/breve), применяемый для моделирования биологических и социально-биологических систем с 3D-визуализацией. Для целей обучения и для решения некоторых практических задач также можно использовать электронные таблицы, лишь бы в них содержалась возможность розыгрыша случайного числа и выбора исходя из условия – *OOo Calc* как раз подойдет.

Концептуальное описание

Начальным технологическим этапом процесса создания и использования имитационной модели является концептуальное описание, включающее постановку целей моделирования, формальное описание объекта моделирования и прочее. Предположим, что на некотором машиностроительном заводе планируется запуск в производство нового продукта. В соответствии с планом, конструкторское бюро (КБ) завода должно осуществить разработку чертёжно-технической документации (ЧТД) в срок 2 месяца. Затем производственные подразделения в течение 1-го месяца должны будут произвести освоение нового продукта и в течение следующих 3-х месяцев произвести опытную партию изделий, которые должны пройти внутренний контроль качества в количестве 10-ти единиц. Целями моделирования определим исследование закономерностей производства новых видов продукции на данном предприятии и оценку затрат времени на разработку и производство нового продукта.

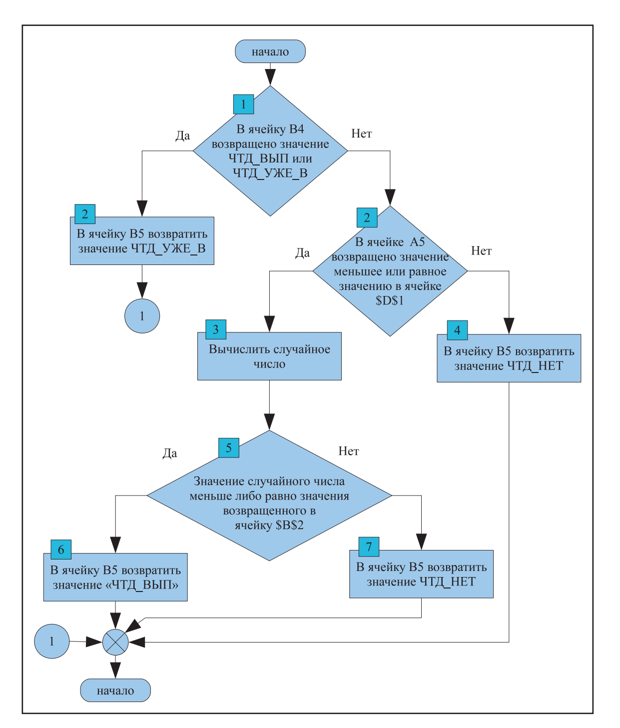
Все остальные условия и допущения будем вводить по ходу создания модели; отметим лишь, что агентами в нашем случае будут выступать подразделения завода, и их поведение будет определяться исходя из двух оценок: компетентности и исполнительности. Под компетентностью будем понимать вероятность того, что результат деятельности того или иного подразделения будет соответствовать некоторому гипотетическому наилучшему, а под исполнительностью – вероятность того, что данное подразделение выполнит возложенное на него задание в установленный срок.

Откуда что берётся?

Резонный вопрос: откуда взять значения компетентности и исполнительности для какого-либо подразделения? Ответствуем прямо: из гипотезы о причинно-следственных связях и факторах деятельности нашего предприятия. Такая гипотеза, будь она формализованной (записанной на бумаге или еще как-нибудь) или просто существующей (порой неосознанно) в умах руководителей и специалистов (экспертов), служит основной предпосылкой для принятия решений. Кстати, посредством моделирования она может быть проверена: если модель при прогонах будет выдавать результаты, схожие с тем, что имеется на самом деле, то гипотеза в некоторой степени верна.

Каждому агенту — свой лист!

Для получения имитационной агентной модели требуется, во-первых, создать агентов, во-вторых, описать правила их взаимодействия. Было бы удобно, чтобы всё связанное с каким-либо агентом помещалось на отдельном листе электронной таблицы. В предварительно открытом и сохранённом документе *OOo Calc* переименуем **Лист 1** (щёлкнуть правой кнопкой мыши, выбрать **Пе-**



› **Рис 1. Блок-схема алгоритма для функции событий разработки ЧТД.**

— всяк за себя

реименовать...): новое название листа пусть будет **КБ**. В ячейке A1 пропишем *Компетентность*, в A2 – *Исполнительность*. Соответственно, ячейки B1 и B2 предназначены для значений компетентности и исполнительности. Предположим, что экспертами для нашего КБ определены значения компетентности **0,88** и исполнительности **0,85**. В ячейке C1 пропишем *Плановый срок*, а в E1 зададим этот самый плановый срок разработки ЧТД в неделях – **8**. Теперь в ячейках A4, B4, C4, E4 соответственно пропишем: *№ раб. недели*, *ЧТД?*, *Качество?*, *Доработка?*. Таким образом, деятельность агента «КБ» будет описываться таблицей состояния со столбцами «№ раб. недели» – это понятно; «ЧТД?» со значением, эквивалентным событиям разработки/не разработки ЧТД; «Качество?» – здесь будет происходить розыгрыш оценки качества ЧТД; «Доработка?» со значениями, эквивалентными событиям доработки ЧТД, если такое понадобится. Заполним первую строку: в ячейке A5 – **1**, первая рабочая неделя, начало проектирования. В ячейку B5 могут возвращаться значения ЧТД_ВЫП, ЧТД_НЕТ и ЧТД_УЖЕ_В; как можно догадаться, первое из них означает, что ЧТД разработано, второе – что не разработано, а ЧТД_УЖЕ_В, так сказать, вспомогательное значение, необходимое для того, чтобы наша модель не выдавала «глупых» результатов, и обозначающее, что событие, эквивалентное разработке ЧТД, случалось раньше. Функция, при помощи которой осуществляется розыгрыш этого события, имеет следующий вид:

```
=IF(OR(B4=”ЧТД_ВЫП”;B4=”ЧТД_УЖЕ_В”); “ЧТД_УЖЕ_В”;
IF((A5>=$D$1);IF(RAND()<=$B$2;
“ЧТД_ВЫП”; “ЧТД_НЕТ”);”ЧТД_НЕТ”))
```

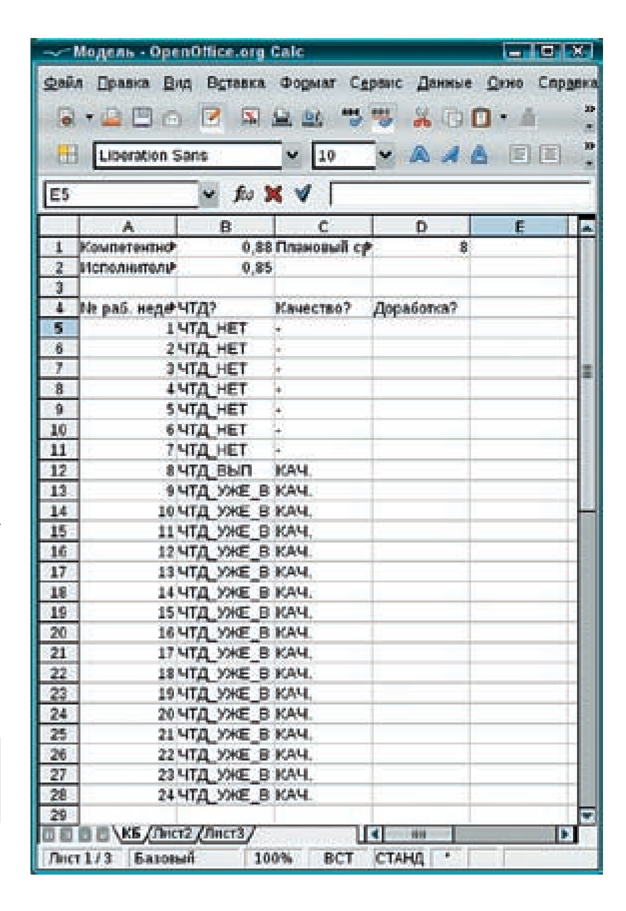
Ее смысл лучше всего раскрыть через блок-схему (см. рис. 1). Если нашему КБ пришел срок отчитываться о проделанной работе и эта работа не была сделана на предыдущей неделе, то с вероятностью, равной исполнительности, разыгрывается значение ЧТД_ВЫП. Функция *OOo Calc* **RAND()** возвращает случайное число в пределах от 0 до 1 с нормальным распределением, и если оно меньше либо равно значению исполнительности, значит, наше КБ справилось с задачей, а в ячейку B5 возвратится значение ЧТД_ВЫП. Обратите внимание на то, что ссылки на некоторые ячейки абсолютные, и на то, что в функциях используются и латинские, и кириллические символы.

Теперь впишем в ячейку C5 функцию, определяющую качество разрабатываемой ЧТД:

```
=IF(D4=”ДОРАБ.”; IF(RAND()<=$B$1;”КАЧ.”; “НЕ_
КАЧ.”);IF(B5=”ЧТД_УЖЕ_В”; C4;
IF(B5=”ЧТД_ВЫП”;IF(RAND()<=$B$1;”КАЧ.”; “НЕ_КАЧ.”); “-”)))
```

Как видно, при различных условиях в ячейку C5 могут быть возвращены три значения: прочерк (или знак «-») для тех недель, когда ЧТД еще не разработано; КАЧ., то есть разработанное ЧТД качественное, и если производственники сделают всё по чертежу, то опытный образец изделия «соберётся»; Не_КАЧ. – соответственно, ЧТД разработано с ошибками.

Осталась еще ячейка D5, это из столбца *Доработка?*. К ней мы вернемся позже, после работы с агентом **Цех**. А пока выделим мышью область ячеек A5:C5, захватим маленький черный квадратик в нижнем правом углу выделения и, удерживая его, распространим наше выделение до 28-й строки (выделенной окажется область A5:C28). Отпустим мышь, и перед нами откроется картина работы КБ на срок 6 месяцев (24 недели, см. рис. 2). Можно уже немножко поэкспериментировать: нажмем F9 несколько раз и увидим, как меняются значения.



› Рис 2. Агент КБ «собственной персоной».

В цехах кипит работа

Приступим к созданию агента, имитирующего поведение производственных подразделений завода. Назовем его **Цех** (переименуйте **Лист 2**). Укажем исходные данные: пропишем в A1 *Компетентность*, соответственно в B1 — **0,80**; в ячейке A2 — *Исполнительность*, в B2 — **0,72**; зададим также C1 — *Плановый срок* »

Нет предела совершенству

Предлагаемая в уроке модель достаточно проста: всего два агента, малое число входных параметров, и зависимости практически очевидны. Для большей полноты отражения реальной системы, в модель можно добавлять агенты, имитирующие другие подразделения, ремонтные и сопутствующие службы, задавать движение денежных средств и материалов, потоков информации. В какой-то момент функции «разрастутся» так, что их будет уже сложно написать правильно. Этим, наверное, возможности электронных таблиц ограничиваются. Хотя, может быть, это лишь вопрос совершенства моделирующего алгоритма.

освоения изделия, в ячейке D1 — **4**; в C2 пропишем *Плановый машинокомплект* — это доля продукции, отведенная для производства в течение планового периода. В нашем случае это **0,83** единицы в неделю; проставим данное значение в ячейке D2.

Дадим названия столбцам таблицы состояния агента **Цех**. Пропишем в A4 *№ раб. недели*; в B4 – *Стадия*; в C4 – *Результат освоения*; в D4 – *Объем производства*; в E4 – *Качество производства*. В A5 пропишем **1** – первая рабочая неделя. В ячейке B5 (столбец «Стадия») введем функцию:

```
=IF(OR(C5=0,25; C5="НЕ_ОСВ"); "ОСВОЕНИЕ";
IF(C5="ОСВОЕНО"; "ПРОИЗВ."; "-"))
```

Если в ячейку в столбце «Результат освоения» той же строки будет возращено или значение 0,25, или значение Не_ОСВ, то в ячейку в колонке «Стадия» будет возвращено значение ОСВОЕНИЕ, то есть наш новый продукт, так сказать, осваивается; если ОСВОЕНО – то ПРОИЗВ., что отмечает переход от стадии освоения к стадии производства, а если ни одно из этих условий не выполняется, то в ячейку строки «Стадия» возвратится прочерк.

Далее, В C5 пропишем

```
=IF(C4="НЕ_ОСВ"; IF(AND(RAND()<=($B$2+(1-$B$2)/2);
RAND()<=($B$1+(1-$B$1)/2); КБ.C4="КАЧ."); 1/$D$1; "НЕ_
ОСВ"); IF(OR(КБ.B4="ЧТД_ВЫП"; КБ.B4="ЧТД_УЖЕ_В");
IF(SUM($C$4:C4)<1; IF(AND(RAND()<=$B$2; RAND()<=$B$1;
КБ.C4="КАЧ."); 1/$D$1; "НЕ_ОСВ"); "ОСВОЕНО"); "-"))
```

Эта функция, в зависимости от различных условий, возвращает значения Не_ОСВ, эквивалентное событию некорректной сборки/производства части изделия; 0,25 – это значение эквивалентно событию, что ¼ нового изделия собрано/произведено, то есть освоено; прочерк, если ЧТД от КБ в производство не поступило. Розыгрыш значений 0,25 и Не_ОСВ начинается в том случае, если у агента КБ в ячейке столбца «ЧТД?» строкой выше содержится значение ЧТД_ВЫП или ЧТД_УЖЕ_В; это эквивалентно тому, что освоение начинается на следующей неделе после того, как будет разработано ЧТД. Значение 0,25 может быть возвращено в том случае, если, во-первых, разработанное ЧТД является качественным, а во-вторых, розыгрыши случайных чисел меньше либо равны значениям компетентности и исполнительности для агента Цех. Если неделей раньше при качественном ЧТД часть изделия не освоена (в ячейке выше возращено Не_ОСВ), то значения компетентности и исполнительности для настоящего момента увеличиваются на величину (1-x)/2, где x – значения вероятности/исполнительности, соответственно. Это логично: у производственных подразделений добавляется опыта. Ну, и если сумма числовых значений в столбце «Результат освоения» становится равным единице, то в ячейку возвращается значение ОСВОЕНО.

Вернемся к агенту КБ, к столбцу «Доработка?». На листе **КБ** в D5 пропишем:

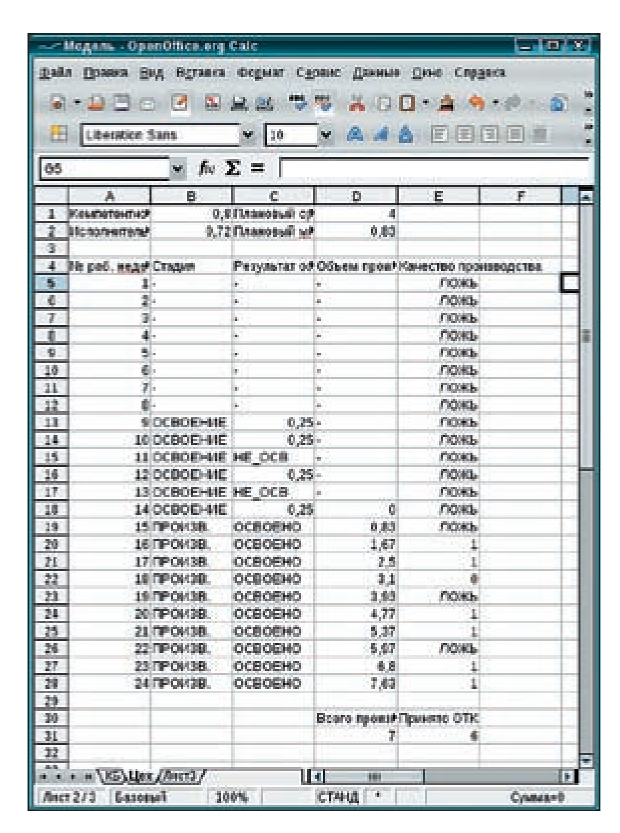
```
=IF(AND(C5="Не_КАЧ."; Цех.C5="НЕ_ОСВ"); "ДОРАБ."; "-")
```

То есть, если разработанное ЧТД некачественное и в цехах «не идет» освоение, то ЧТД дорабатывается: в ячейке строкой ниже в столбце «Качество?» с вероятностью, равной компетентности КБ, будет производиться розыгрыш значения качества ЧТД. Не забудем вставить эту функцию в каждую ячейку столбца: выделим с помощью Shift область D5:D28 и нажмем Ctrl+D.

Переключимся на лист **Цех**: в D5 (столбец «Объем производства») пропишем

```
=IF(AND(B5="ОСВОЕНИЕ"; B6="ПРОИЗВ."); 0; IF(B5="ПРОИЗВ.";
IF(RAND()<=$B$2; D4+$D$2; D4+$D$2*$B$2); "-"))
```

Эта функция рассчитывает объем произведенной продукции нарастающим итогом. Если розыгрыш случайного числа меньше либо равен значению исполнительности производственных подразделений, то к произведенному ранее объему (вначале это ноль) добавляется значение планового машинокомплекта; если розыгрыш случайного числа больше значения исполнительности,



	A	B	C	D	E	F
1	Компетентн	0,8	Плановый ср	4		
2	Исполнитель	0,72	Плановый м	0,83		
3						
4	№ раб. неде	Стадия	Результат ос	Объем прои	Качество производства	
5	1	-	-	-	ЛОЖЬ	
6	2	-	-	-	ЛОЖЬ	
7	3	-	-	-	ЛОЖЬ	
8	4	-	-	-	ЛОЖЬ	
9	5	-	-	-	ЛОЖЬ	
10	6	-	-	-	ЛОЖЬ	
11	7	-	-	-	ЛОЖЬ	
12	8	-	-	-	ЛОЖЬ	
13	9	ОСВОЕНИЕ	0,25	-	ЛОЖЬ	
14	10	ОСВОЕНИЕ	0,25	-	ЛОЖЬ	
15	11	ОСВОЕНИЕ	НЕ_ОСВ	-	ЛОЖЬ	
16	12	ОСВОЕНИЕ	0,25	-	ЛОЖЬ	
17	13	ОСВОЕНИЕ	НЕ_ОСВ	-	ЛОЖЬ	
18	14	ОСВОЕНИЕ	0,25	0	ЛОЖЬ	
19	15	ПРОИЗВ.	ОСВОЕНО	0,83	ЛОЖЬ	
20	16	ПРОИЗВ.	ОСВОЕНО	1,67	1	
21	17	ПРОИЗВ.	ОСВОЕНО	2,5	1	
22	18	ПРОИЗВ.	ОСВОЕНО	3,1	0	
23	19	ПРОИЗВ.	ОСВОЕНО	3,93	ЛОЖЬ	
24	20	ПРОИЗВ.	ОСВОЕНО	4,77	1	
25	21	ПРОИЗВ.	ОСВОЕНО	5,37	1	
26	22	ПРОИЗВ.	ОСВОЕНО	5,97	ЛОЖЬ	
27	23	ПРОИЗВ.	ОСВОЕНО	6,8	1	
28	24	ПРОИЗВ.	ОСВОЕНО	7,63	1	
29						
30				Всего произ	Принято ОТК	
31				7	6	
32						

Рис 3. Разработка модели завершена.

то к произведенному ранее объему добавляется значение планового машинокомплекта, умноженного на значение исполнительности. Иными словами, в нашей модели объем производства на текущей неделе может принимать только два значения.

Зададим функцию, определяющую, какое количество из произведенных изделий является качественным. В E5 пропишем:

```
=IF(B5="ПРОИЗВ."; IF((FLOOR(D5;1)-FLOOR(D4;1))=1;
IF(RAND()<=$B$1; 1; 0)))
```

Как только разность между округлёнными вниз значениями из соседних ячеек столбца «Объем производства» станет равна единице, что эквивалентно появлению завершенного изделия, определяется его качество: если розыгрыш случайного числа меньше либо равен компетентности, то изделие качественно, и в ячейку возвратится значения 1, а если нет, то 0.

Выделим A5:E5, захватим мышью маленький черный квадратик в правом нижем углу и распространим выделение на область A5:E28. Ну вот, агент Цех практически готов. Добавим лишь подсчет количества произведенной продукции и продукции принятой ОТК, в D30 пропишем *Всего произведено*, в D31 –

```
=FLOOR(D28;1)
```

В E30 введем *Принято ОТК*, в E31 –

```
=SUM(E5:E28)
```

Наша модель готова (см. рис. 3).

Вычислительный эксперимент

После того, как имитационная модель реализована в программе, должна быть проведена её комплексная проверка с целью оценки достоверности по отношению к реально существующей системе. Это этап отладки и исследования свойств модели. Мы допустим, что всё это сделано, и сразу перейдем к направленному вычислительному эксперименту. Переименуем **Лист 3** в **Статистика**. В ячейках A1, B1, C1, D1 соответственно введем: *ЧТД_ВЫП?*, *ОСВОЕНИЕ?*, *Всего произведено*, *Принято ОТК*. В A2 (это под *ЧТД_ВЫП?*) пропишем функцию:

```
=MATCH("ЧТД_ВЫП";КБ.B5:B28; 0)
```

Она возвратит номер рабочей недели, когда «сделают» ЧТД. В ячейке B2 (под *ОСВОЕНИЕ?*) зададим функцию, которая будет вычислять количество недель, затраченных производственниками на освоение:

```
=MATCH(“ПРОИЗВ.”; Цех.B5:B28; 0)-MATCH(“ОСВОЕНИЕ”;Цех.B5:B28; 0)
```

В C2 и D2 выведем значения целого количества произведенных единиц продукции и количество единиц продукции, принятых ОТК. На листе **Цех** мы уже рассчитывали эти величины, поэтому просто сошлемся на соответствующие ячейки. Для C2 –

```
=Цех.D31
```

а для D2 –

```
=Цех.E31
```

Теперь выберем меню Сервис > Макросы > Записать макрос. Затем выделим диапазон A2:D2, нажмем Ctrl+C, Ctrl+Стрелка вниз, Ctrl+Стрелка влево, Стрелка вверх, далее Ctrl+V, в открывшимся окне выберем **Числа** и **Форматы**, нажмем OK, завершим запись макроса. В появившемся диалоге управления макросами дадим ему название **Statistic**. Снова перейдем в меню Сервис > Макросы > Управление макросами > OpenOffice.org Бейсик..., нажмем Редактировать. В открывшемся документе перед **sub Statistic** пропишем:

```
Sub Main
For i=1 to cLng(inputbox(“число прогонов”, “прогонов будет...”, “10”))
Statistic
Next
End Sub
```

и нажмем Ctrl+S. Теперь при выполнении макроса **Main** результаты прогонов будут записываться на листе «снизу вверх»; для *OOo Calc 2.4* это с 65535-й строки. Можно ограничить такие «прыжки» заранее, проставив результаты первого прогона в нужной строке через Ctrl+V.

Определение количества прогонов, необходимого для того, чтобы результаты, полученные на имитационной модели, были статистически значимы, происходит на этапе планирования вычислительного эксперимента. Это важный момент, на котором также определяются цели направленного вычислительного эксперимента, методы сбора и обработки результатов. В соответствии с нашим первоначальным заявлением, целью вычислительного эксперимента будет оценка выходных переменных при заданных параметрах системы.

Метод обработки результатов будет заключаться в экспертном заключении на основе статистического анализа выходов. Для определения количества прогонов есть общая формула: $N=pq^k$, где p – число повторений, q – число уровней входов, k – число входов. На практике также пользуются формулой $N \approx N_0 * d_0^2/d^2$, где N_0 – количество пробных прогонов, d_0 – доверительный интервал, определенный по результатам пробных прогонов, d – требуемая точность. Выделим A2:D2, нажмем Ctrl+C, перейдем к A12, нажмем Ctrl+V, в появившемся окне выберем Числа и Форматы, нажмем OK. Запустим макрос **Main**: меню Сервис > Макросы > Управление макросами > OpenOffice.org Бейсик... > Выполнить. В появившемся окне (рис. 4) поставим **9**, нажмем OK – и увидим, как заполнятся ячейки значениями результатов пробных прогонов (всего их будет десять). Одним из основных интересующих нас выходов модели является значение принятых ОТК изделий; в этой связи определим доверительный интервал для данной величины при уровне α=0,05 и в G4 пропишем:

```
=CONFIDENCE(0,05; STDEV(D5:D14); 10)
```

Смысл полученной величины d_0 заключается в том, что математическое ожидание количества принятых ОТК изделий с вероятностью 1-α=0,95 будет находится в пределах $x_{ср} \pm d_0$, где $x_{ср}$ – среднее значение выборки. Точность d – это задаваемая величина, в пределах которой значение $x_{ср}$ может отличаться от математического ожидания. При десяти пробных прогонах d_0 получилось равным 0,83 (при других пробных прогонах это значение может быть иным); такая точность для нашего случая вполне удовлетворительна, так что этим можно и ограничиться. Вычислим средние значения и доверительные интервалы выходов модели, в A16 пропишем *Средние значения*; в ячейке A17 –

```
=AVERAGE(A5:A14)
```

Распространим эту функцию на B19:D19 известным нам способом. В A18 пропишем *Доверительные интервалы*, в A19 –

```
=CONFIDENCE(0,05; STDEV(A5:A14); 10)
```

То же сделаем для B21:D21.

Ну вот, теперь осмотрим всё (рис. 5) и оценим. Выходит (если гипотеза о причинно-следственных связях и факторах верна), что с первоначальными планами наша система, скорее всего, не справится, и необходим дальнейший анализ и разработка мероприятий по увеличению компетентности и исполнительности – прежде всего, для производственных подразделений. К слову сказать, наша модель близка к реальной ситуации на очень многих отечественных машиностроительных предприятиях, так что полученные результаты – это лишний повод для размышлений. **LXF**

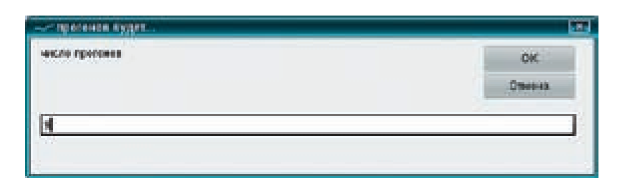


Рис. 4. Задаем количество прогонов.

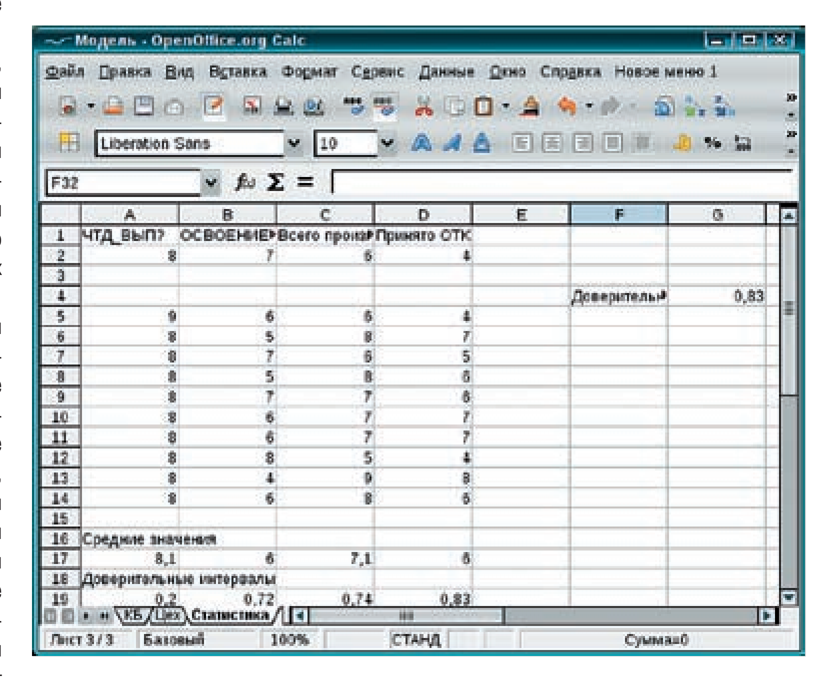


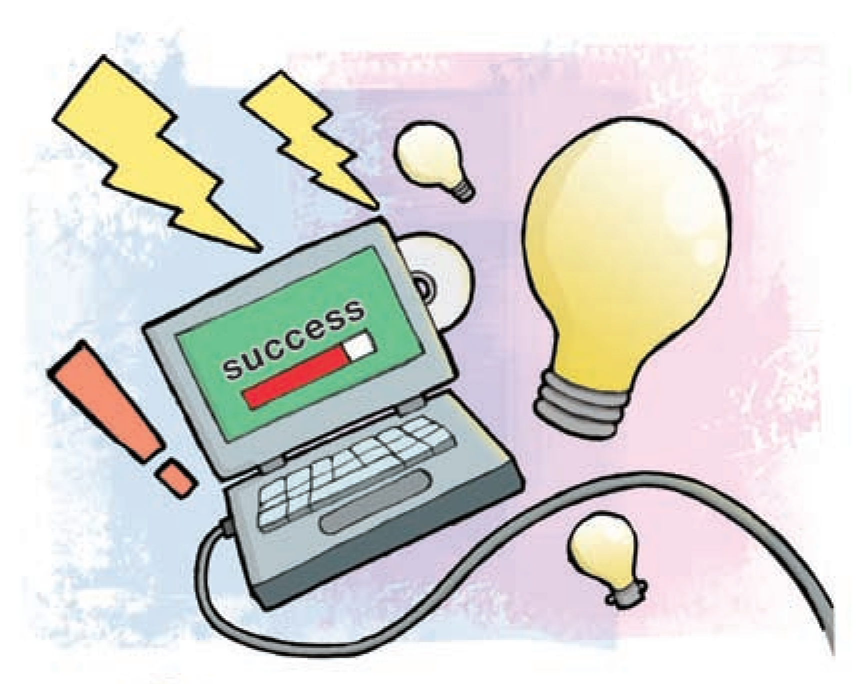
Рис 5. Результаты прогонов и статистика.

Если возникают вопросы

При решении в *OOo* различных задач могут возникнуть вопросы, которые сложно решить, пользуясь одной лишь встроенной справкой. На счастье, есть куда обратиться: в сети существуют и вики, и FAQ, и форумы, где обсудят вашу проблему и предложат варианты её решения – например, сообщество на http://community.i-rs.ru.

SELinux: Страж

Боб Мосс научит, как прикрыть вашу Linux-систему, не добавляя мер безопасности, препятствующих выполнению повседневных задач.





Наш эксперт

Боб Мосс делит свое время между программированием и обучением на факультете вычислительной техники.

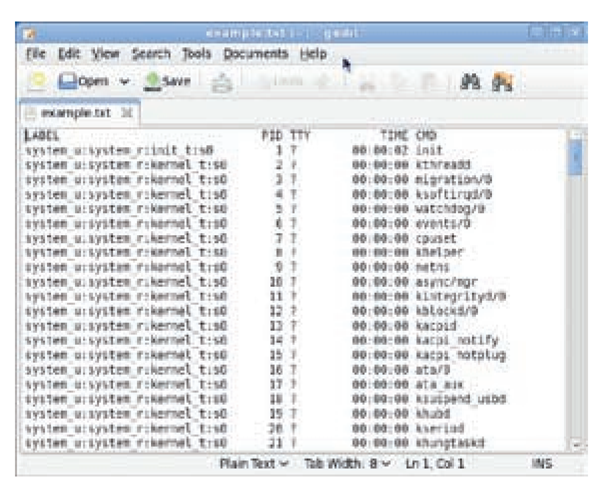
Спешу успокоить тех, кто ранее не слышал о SELinux [Security Enhanced Linux – Linux с повышенной безопасностью]: вы не одиноки. SELinux идет в составе большинства ведущих дистрибутивов, но по умолчанию выключен (а есть и пользователи, отключающие его намеренно); в итоге этот мощный инструмент безопасности слабо востребован и полузабыт.

Любопытствуете, почему так случилось? Начнем с самого начала: когда US National Security Agency [Агентство национальной безопасности США] открыло его под свободной лицензией, SELinux влился в основной код ядра. Однако он загораживал дорогу людям, занятым в своих системах Red Hat выполнением ежедневных заданий, и был слишком сложен и громоздок для настройки пользователями среднего уровня, так что они обычно просто сдавались и отключали его.

Тем не менее с тех пор многое изменилось, и у него даже возникли конкуренты (например, *AppArmor*), предоставляющие такой же высокий уровень безопасности для стандартного настольного компьютера с Linux.

Так в чем же идея SELinux? По сути, он присваивает всем пользователям, процессам и файлам категории, а затем предоставляет этим «объектам» самый минимум прав, которые им нужны для выполнения своих функций. Если объект пытается выйти за эти рамки или вторгается в действия другого объекта, его останавливают и сообщают вам об этом в соответствии с заданной политикой безопасности.

SELinux, однако, поставляется с набором правил, покрывающих только типовые сценарии работы, и в один прекрасный день, приступая к очередной важной задаче, вы можете обнаружить, что SELinux взбрыкнул и остановил вас.



› **Ваши политики не ограничены файлами: они влияют и на запущенные процессы, и на пользователей.**

Но прежде чем в ярости швырять свой компьютер за окно или немедленно вырубать SELinux, просто выдайте себе права, позволяющие делать то, что вам надо, и там, где надо – это не слишком трудно. Со временем ваше терпение окупится: ведь укрепится безопасность вашей машины. На данном уроке мы разберемся, как работают политики безопасности SELinux, как изменить текущий режим SELinux для конкретных сценариев и как проверить выданные вам права, когда вы приспособите SELinux для своих личных компьютерных нужд.

Первый шаг

Первым делом надо убедиться, установлен ли SELinux в вашей системе. Пользователи Fedora скорее всего заметят, что он присутствует по умолчанию, благодаря родству с Red Hat – просто наберите следующие строки в стандартном терминале пользователя, чтобы увидеть, в каком состоянии SELinux находится сейчас:

```
getenforce
```

Вы можете получить один из трех результатов: **enforcing**, **permissive** или **disabled**. Первое говорит, что SELinux принуждает к выполнению всех заданных политик безопасности и сообщает обо всех нарушениях. **Permissive** показывает, что SELinux сообщает обо всех нарушениях, но насильно заданные правила безопасности не применяет; а последний вариант – SELinux отключен.

Пользователи Ubuntu нуждаются в нескольких добавочных шагах для установки SELinux, поскольку по умолчанию обычно установлен *AppArmor*. Прочитайте врезку «*Альтернатива – AppArmor*» на соседней странице, и если все же решите держаться SELinux, установите его через ваш менеджер пакетов. Покончив с этим, наберите в терминале следующую команду, для его запуска:

```
sudo sed -i 's/SELINUX=.*/SELINUX=enforcing/' /etc/selinux/config
```

Она изменит состояние SELinux прямо через его конфигурационный файл. Вы также можете изменить **enforcing** на **permissive**

вашей системы

Альтернатива — AppArmor

Пользователи Ubuntu или OpenSUSE могут обнаружить, что *AppArmor* уже установлен по умолчанию, хотя последние должны включать его вручную (как показано на экранном снимке). Те, кто знаком с SELinux, могут найти некоторые аспекты *AppArmor* очень похожими. Например, *AppArmor* имеет режимы Enforce, Complain и Disabled – полная аналогия с SELinux. Его также чуть легче запустить и остановить: пользователи Ubuntu могут просто набрать

```
rcapparmor start
```

Замена **start** на **stop** даст обратный эффект. Вы можете также использовать **reload** для перезагрузки всех текущих профилей *AppArmor* (эквивалента правил SELinux), или **status** для отображения количества загруженных профилей в режимах Complain и Enforce. Регистрацией событий в журнале также легко манипулировать:

```
rcaaeventd start
```

Как и ранее, для обратного эффекта можно просто использовать **stop** вместо **start**.

Преимуществом *AppArmor* является его режим Learning, который наблюдает, как ваша машина используется в заданный вами промежуток времени, и затем применяет эту информацию для выяснения того, что разрешать делать, а что нет.

Трещины в броне

Это чрезвычайно полезно для настройки web-серверов, где локальный доступ в идеале должен быть ограничен, а web-страницы должны быть доступны только особым людям по определенным путям.

Звучит такая идея замечательно, но на практике имеет препятствие. В режиме Learning *AppArmor* легко перенимает дурные привычки, то есть видит события, которые вы бы сочли нарушениями, но не предоставляет доклада об этом, считая такое поведение нормальным. Это может быть смягчено осторожностью поведения и в некоторых сценариях особенно не повредит, но все-таки должно считаться недостатком. У SELinux такой проблемы нет, потому что он распространяется с набором политик безопасности для основных востребованных сценариев, и затем позволяет пользователю при необходимости задать более точные политики. Конечно, потребуется больше усилий и некоторый уровень понимания, но зато вы узнаете о всех нежелательных событиях на вашей машине сразу при их появлении.



Оба решения имеют свои «за» и «против», но после нашего урока вы сможете сделать более обоснованный выбор, что именно ставить на ваш любимый дистрибутив.

или **disabled**, в соответствии с текущим режимом вашей деятельности.

У пользователей Fedora маршрут для включения SELinux и изменения режима работы много проще. Одна из возможностей – применение инструмента с графическим интерфейсом из System > Administration [Система > Администрирование] рабочего стола Gnome: изменения вносятся через выпадающее меню, после чего остается только кликнуть Apply [Применить]. Если вам больше нравится командная строка – то же самое действие выполнится при наборе

```
echo 0 > /selinux/enforce
```

Оператор **0** велит SELinux выключить режим Enforce до окончания текущей сессии и перейти в режим Permissive. Если вы захотите вернуть режим Enforce, просто запустите эту же команду снова, но установив **1** вместо **0**.

> «Ваше терпение окупится: укрепится безопасность машины.»

При желании изменить режим насовсем, откройте от имени root файл **/etc/selinux/config** в вашем текстовом редакторе и исправьте соответствующую строку **SELINUX**:

```
SELINUX=enforcing
```

Опять же, можете изменить **enforcing** на **permissive** или **disabled**, как вам удобнее.

Тут вас, видимо, посетит вопрос: а зачем вообще менять режимы работы SELinux? Суть в том, что при выполнении любой административной задачи или решении проблемы может оказаться, что политики SELinux (рассчитанные на повседневное использование на настольных компьютерах) мешают вам работать. То есть в данном случае отключение полезно; но когда вы снова будете использовать ваш компьютер как обычно, вы должны иметь возможность заставить настроенную конфигурацию SELinux тихо работать в фоновом режиме, не досаждая вам.

Позвольте вам не позволить

При запущенном SELinux иногда обнаруживается, что вещи, работавшие ранее, перестали это делать (или вы видите диалоговое окно, угрожающее сообщить обо всем вам же). В этом случае можно попробовать полезный инструмент *Audit2allow*: он соберет данные о недавних действиях, которые предотвратил SELinux, и позволит вам разрешить их. Перед его использованием проверьте, что у вас запущен демон *auditd*, посредством команды

```
ps -ef | grep auditd
```

Если вы видите **auditd** в списке, и перед ним нет слова **grep**, значит, демон запущен. Не паникуйте, если он не появился, потому что единственное различие нижеприведенной команды будет в замене **a** на **d**:

```
audit2allow -l -a -M local
```

Параметр **-l** читает запреты с момента последней перезагрузки политик (в большинстве случаев – с момента вашего входа в систему), а **-m** перемещает запреты в модуль, который вы можете употребить для составления правил. Учтите, что если вы хотите использовать эту строку более одного раза, имя **local** нужно будет изменить, во избежание конфликтов. Удовлетворившись правилом, соберите его – для этого достаточно набрать **semodule -i local.pp**.

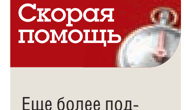
SELinux: Свои

Научившись применять SELinux, почему бы не создать правила самим?

Теперь вы можете настроить SELinux и знаете, как создать новое правило, если система безопасности начнет мешать выполнению повседневных задач на вашей Linux-машине. Однако в вашей системе бывают такие файлы и зоны, которые желательно уберечь от чужих глаз или защитить дополнительно. Тогда нужно создать собственные правила.

Опять же, от правил мало проку, если нет возможности проверить политики SELinux, применяемые в той или иной ситуации, и этот вопрос мы также рассмотрим.

Итак, вы убедились, что SELinux присутствует и запущен в правильном режиме работы, и изучили, как обойти его блокировки для допустимых действий; следующим шагом приступайте к применению ваших собственноручно заданных правил. При желании можно просто использовать встроенный в Fedora мастер правил, но уж если вы зашли так далеко, то, скорее всего, захотите знать больше о том, что происходит за кулисами безопасности.



Еще более подробную информацию о SELinux вы получите на http://selinuxproject.org.

Сперва рассмотрим, что SELinux понимает под 'user' [пользователь]. Это не то же самое, что обычный пользователь системы, который может повысить привилегии до root с помощью **sudo** или **su**. На самом деле, многие системные пользователи имеют «двойников» в SELinux, поэтому удобнее считать пользователей SELinux подвидом «группы пользователей», каждая из которых может содержать любое количество пользователей системы с одинаковыми ролями и разрешениями. Когда пользователь SELinux упоминается в правиле или системной политике, к его имени обычно приписывается окончание **_u** (к примеру, **bob_u**).

Ну, а что такое роль пользователя? Пользователь SELinux может иметь их столько, сколько вы для него зададите, и каждая роль определяется политикой SELinux. Примерами ролей могут быть «посетитель» или «администратор сайта». Пользователи объектов, ассоциированных с ролями, по соглашению обычно имеют в конце имени **_r**, поэтому при просмотре политики их можно относительно быстро засечь.

> «Бывают зоны, которые желательно защитить дополнительно.»

Каждое правило должно также ссылаться на 'type' [тип] (в Интернете его иногда называют «домен»), определяя, какие ресурсы разрешены к доступу, а какие нет. Опять же согласно правилам, имя типа оканчивается на **_t**.

Следующим шагом будет создать 'context' [контекст] для правила, вооружившись этими тремя понятиями. Это по сути означает помещение пользователя, роли и типа вместе в первой половине правила; но можно использовать их и по отдельности.



› Fedora содержит мастер, позволяющий создать ваши политики безопасности быстро и просто.

Сохраним это в контексте

По желанию вы можете добавить в этот контекст 'range' [диапазон], но это не является строго необходимым, пока вы не применяете правила для множества различных типов. Ваш контекст может иметь следующую форму:

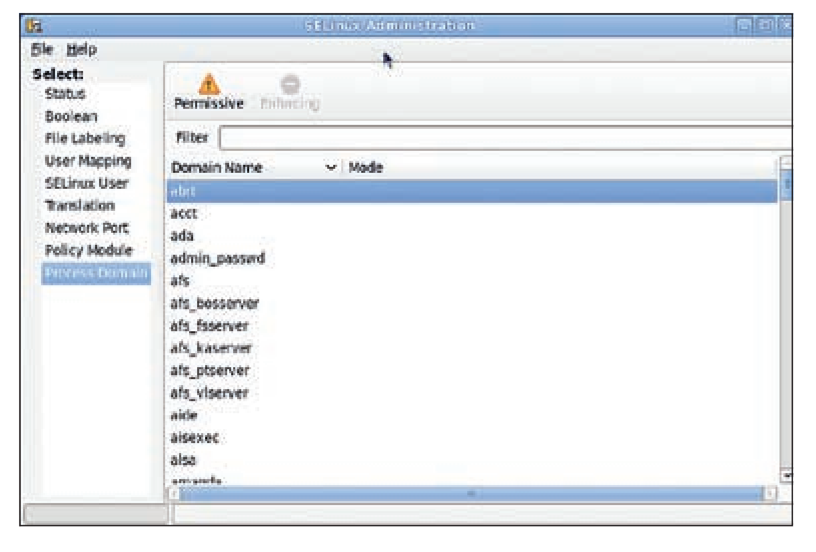
```
user:role:type:range
```

А контекст без диапазона был бы

```
system_u:system_r:xserver_t
```

В данном контексте вы ссылаетесь на пользователя **system** с ролью **system** и привилегиями доступа типа **xserver** (вполне конкретная вещь). Если бы здесь не было двоеточий, вы бы задали правила для всех пользователей **system**, всех пользователей с ролью **system** и всех пользователей с привилегиями доступа как у **xserver_t**; этим условиям, конечно, удовлетворяет гораздо больше пользователей.

Теперь, получив представления о том, что такое 'context', вы можете применить свой контекст к любому классу объектов и дать им подходящие права. SELinux предоставляет мириады классов объектов (лучше сказать, категорий объектов) для поль-



› Тем, кто побаивается командной строки, управлять SELinux поможет графический интерфейс.

» Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.

правила лучше

зователей, процессов и файлов. Примерами классов объектов будут **dir** для директорий и **file** для файлов.

Далее, каждый из этих объектов имеет набор прав, которые вы можете выдавать или отбирать. Например, **file** может иметь права на создание, разлинковку (удаление), чтение и запись, а объект **network** – на подключение и передачу данных. Исчерпывающий список всех доступных классов объектов и их ассоциированных прав имеется на сайте проекта SELinux: http://bit.ly/9idu42.

Применяем правила

К примеру, выдача ваших прав классов объектов двум различным типам пользователей будет выглядеть так

```
allow user_t user_home_t:file { create read write unlink };
```

Как видите, **allow** используется для разрешения данным конкретным типам пользователей выполнять перечисленные действия над файлами (а для запрещения команда будет **deny**).

Вы можете применять похожие правила к типам, именам или комбинациям всех трех, о чем мы говорили выше, в зависимости от того, что вы хотите позволить или запретить в вашей системе для определенных объектов; так у вас будет максимальный контроль над тем, как работает ваша машина.

Теперь, когда вы разбираетесь в правилах, должно быть ясно, что глобально насаждаемые системные настройки могут в некоторых ситуациях не работать. К примеру, вам может понадобиться временно изменить часть вашей системы с последующим возвратом умолчаний SELinux.

Вместо глобального изменения режима работы SELinux, что влияет на всю систему, можно просто сменить режим работы отдельного правила, используя **semanage**:

```
semanage permissive -a myapp_t
```

Попробуйте заменить **myapp_t** на любой другой объект или контекст. Затем просто замените **a** на **d**, чтобы вернуть данный объект назад к системным умолчаниям.

Покончив с манипуляциями над SELinux, заставляющими его работать как вам хочется, вы можете обнаружить, что нужно излечить какую-то проблему или вам просто интересно узнать, что делает каждая политика SELinux в том или ином случае. Это особенно удобно, если вы пытаетесь передать файл или установить «нехороший» пакет, от которых SELinux вас защищает. Есть две строки, применяемые в терминале, и в обеих фигурирует магическое и регистрозависимое **-Z**, о чем упомянуто на множестве сайтов в Интернете:

```
ls -alZ *
```

Эта строка выведет список всех файлов в текущей директории с их ассоциированным владельцем, правами и правилами SELinux (просто замените * на имя файла или нужный путь). Похожее действие для процессов, в настоящий момент работающих в системе, выполняет следующая строка:

```
ps -eZ > example.txt
```

Она выведет список всех запущенных процессов вместе с их ассоциированными правилами SELinux и затем вставит эту информацию в текстовый файл, чтобы вы его проанализировали. Отобразить все эти данные в окне терминала тоже можно, но это менее удобно: результат будет некрасивый и более сложный для последующей интерпретации.

И, наконец, если нужно проверить, какие активные пользователи системы имеют ассоциированные политики SELinux, просто наберите **id -Z** в вашем терминале.

Итак, вооружившись чуточкой терпения и этими инструкциями, можете приступать к своим повседневным задачам, в полной уверенности, что ваш компьютер хорошо защищен. LXF

```
bob@localhost:/media/VBOXADDITIONS_3.1.2_56127
File  Edit  View  Terminal  Help
[bob@localhost VBOXADDITIONS_3.1.2_56127]$ ls -alZ *
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   AUTORUN.INF
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   autorun.sh
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   VBoxLinuxAdditions-amd64.ru
n
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   VBoxLinuxAdditions-x86.run
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   VBoxSolarisAdditions.pkg
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   VBoxWindowsAdditions-amd64.
exe
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   VBoxWindowsAdditions.exe
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   VBoxWindowsAdditions-x86.ex
e

32Bit:
dr-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   .
dr-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   ..
dr-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   OS2
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   Readme.txt

64Bit:
dr-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   .
dr-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   ..
-r-xr-xr-x. bob bob system_u:object_r:iso9660_t:s0   Readme.txt
[bob@localhost VBOXADDITIONS_3.1.2_56127]$
```

› **Можно применить защитные политики SELinux ко всем своим файлам или создать собственное правило, которое будет работать в конкретных случаях.**

Любителям приключений

Тем, кто хочет добраться до реальной начинки работающего механизма SELinux или писать и собирать собственные политики безопасности без помощи графического интерфейса или инструментов командной строки, будет полезно узнать, как SELinux размещается в корневом разделе.

Все основные файлы конфигурации, используемые и модифицируемые сопутствующими приложениями и утилитами, доступны в **/etc/selinux**. Файлы, относящиеся к политикам, находятся в том же каталоге, имеют соответственные имена и снабжены поддиректорией **modules**, где расположены дополнительные настройки политик. Вы также можете найти имена политик в файле **/etc/selinux/config**, начиная со строки **SELINUXTYPE=**.

Модули и файлы SELinux, зависящие от ядра системы, расположены в директории **/selinux** и содержат текущую конфигурацию и активные политики безопасности. Если вы не абсолютно уверены в своих действиях, лучше держаться подальше от этой директории, потому что последствия ошибок будут катастрофическими – вы можете поломать систему или, чего доброго, заблокировать доступ себе же!

Написание и компиляция своих собственных политик SELinux без инструментов выходит за рамки данного учебника, но на сайте Red Hat опубликован учебник, охватывающий этот высший пилотаж. Но, повторюсь, это для продвинутых пользователей, о чем четко указано по адресу http://bit.ly/cIXESW.

Python: Потоки

Скомбинируйте мощь видео от *GStreamer* с графической изощренностью *Clutter* – получится то, что **Ник Вейч** называет «чистая работа».





Наш эксперт

Ник Вейч великий укротитель питонов категории A++++. Мы, безусловно, будем обращаться к нему снова.

Первичная задача *Clutter*'а – упрощение процесса создания приятных графических интерфейсов. Он применяет подход «выстрелил-и-забыл» ко множеству вещей – в частности, к анимации: задав последовательность, запустите ее, и пусть себе выполняется.

Однако самому *Clutter* со всем этим не справиться. Анимация в главном модуле превосходна, но в других областях есть изъяны. Начнем с того, что в *Clutter* есть всего несколько примитивных объектов-актеров; поэтому к нему и прилагается расширение на базе *Cairo* (на следующем уроке мы рассмотрим его подробнее).

Другая библиотека, на которую опирается *Clutter* – это *GStreamer*, кудесник по части потоков медиа-данных. Когда речь заходит о мультимедиа, *GStreamer* – одна из наиболее часто используемых библиотек, и неудивительно, что для нее есть модуль Python. Чтобы применять его в своих приложениях, нужно немного разобраться в инструментарии *GStreamer*, но здесь, как всегда, на это мало места. К счастью, он не сложен. Нам и не нужно ничего замысловатого – есть у нас URI видеофайла или какого-нибудь другого источника, вот мы и хотим его воспроизвести.

Сначала зададим обычную сцену [stage] в *Clutter*:

```
>>> import gst
>>> import clutter
>>> import cluttergst
>>> stage=clutter.Stage()
>>> stage.set_title("Clutter_Streamer")
>>> stage.set_size(320,290)
>>> stage.set_color(clutter.Color(255,255,255,255) )
>>> stage.show()
>>>
```

Если вы в курсе предыдущих эпизодов нашего учебника, вам это должно показаться знакомым. Для новичков скажем, что здесь импортируются самые важные модули (заметьте, что наряду с **gst** нам понадобится **cluttergst**), создается объект Stage (окно, на языке *Clutter*), ему дается заголовок, размер и цвет фона, а затем он отображается. Вы увидите белое окно. Если нет, проверьте синтаксис, или скопируйте файл **listing** с DVD.

Весь мир — сцена

Теперь начнется интересное. Объекты, отображаемые на сцене *Clutter*, называются актерами [actor]. Существует несколько типов актеров. Некоторые относятся к базовым, например, объекты «прямоугольник» и «текст», и живут в главном модуле *Clutter*, но есть и другие, специальные актеры из внешних библиотек.

Объект **VideoTexture** находится в модуле **cluttergst**, и он похож на актера **rectangle**. У него много свойств из тех, что мы исследовали ранее, но он особый, так как подцепляется к проигрывателю видео и использует его для обновления своей текстуры. Это одна



› **Мы еще не написали ни строки кода, а в окне уже вовсю идет потоковое видео.**

» **Месяц назад** Мы создали анимированную читалку для новостей RSS.

видео



из сильных сторон *Clutter* – задав актера или даже установив ему анимацию на выполнение, незачем возвращаться к нему для контроля: он продолжит работать и делать то, что ему велено.

Проигрыватель видео известен под названием playbin; это встраиваемый плейер. Вы задаете ему ресурс для воспроизведения, и он увязывает все остальное самостоятельно. Давайте так и поступим:

```
>>> vid=cluttergst.VideoTexture()
>>> playbin=vid.get_playbin()
>>> playbin.set_property('uri','mmsh://live.camstreams.com/
cscamglobal16MSWMExt=.asf')
>>>
```

Сначала мы задали актера текстуры видео, затем извлекли созданный объект playbin. Прежде чем воспроизводить что-либо, укажем источник. Лучший способ сделать это – вызвать метод **set_property** объекта playbin, включающий свойство, которое мы хотим задать (URI), и то, что мы хотим ему присвоить. Здесь мы использовали web-адрес видео в Интернете: камера отслеживает дорожное движение на площади Пикадилли в Лондоне и дает поток живого видео (в отличие от тех, что просто обновляют картинку каждые пять секунд). Вам сразу же захочется избежать головокружительного восторга от вида двух дюжин автобусов, бодающихся перед светофором, и использовать локальный файл. Это значит, что все в порядке.

URI может ссылаться на файл или вообще что угодно, распознаваемое как корректный источник. *GStreamer* определит, что это такое и что с этим делать, при наличии правильных модулей расширения. Если таковых нет, возникнет сообщение об ошибке, говорящее о невозможности воспроизвести поток. Возьмите что-нибудь попроще: например, файл видео, который вы уже успешно просмотрели в *Totem*.

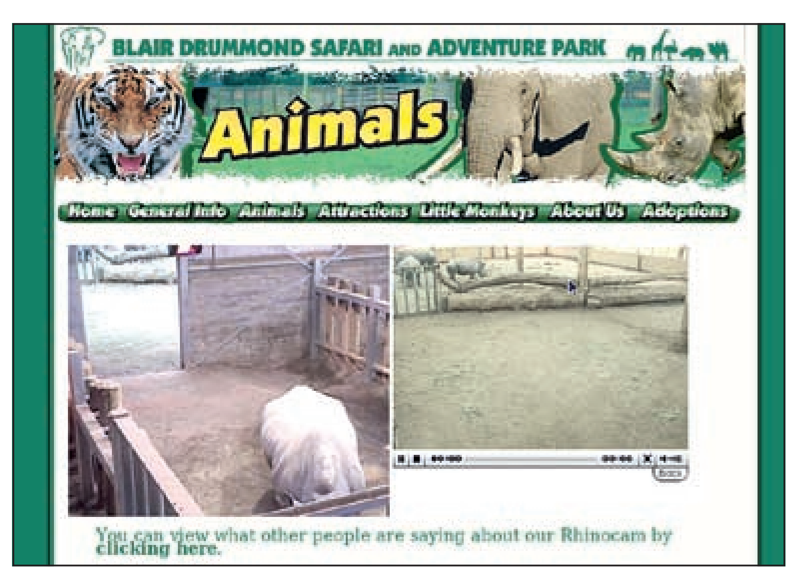
Для URI файлов (или чего угодно) потребуется префикс подходящего протокола. Например, правильным URI для локального файла будет **file:///home/evilnick/Videos/killmike.ogg**.

Затем создадим другой объект *GStreamer*, чтобы использовать его для управления ранее созданным объектом playbin:

```
>>> pipe=gst.Pipeline("pipe")
>>> pipe.add(playbin)
>>> stage.add(vid)
```

Мой поток отказал

Если канал *GStreamer*'а закрывается с ошибкой, или, реже, просто ничего не делает – проблема, возможно, с кодеком. Недостающие в вашем дистрибутиве кодеки можно добавить, установив пакет под названием **gstreamer_plugins_good** или наподобие. Все прочие пакеты, видимо, «**bad**» или «**ugly**» (здесь тонкий отсыл к мировой культуре, если кто понял). В зависимости от вашего местожительства, вы сможете законно скачать и установить дополнительные модули, чтобы увидеть больше потоков.



Ух ты! Поискав, вы найдете немало любопытных web-камер. Или возьмите свои файлы.

Создав объект pipe, мы подключили его к playbin, с помощью pipe'овского метода **add()**. pipe используется просто как контейнер для нашего плейера. Наконец, на сцену можно добавить объект текстуры видео – именно он обновляется при воспроизведении потока.

Теперь у нас есть белое окно с отображаемой в нем текстурой видео, но посмотреть его пока нельзя: ведь оно еще не включено.

```
>>> pipe.set_state(gst.STATE_PLAYING)
<enum GST_STATE_CHANGE_ASYNC of type
GstStateChangeReturn>
```

Контейнер pipe используется для управления потоком. С помощью предопределенной в **gst** константы **gst.STATE_PLAYING** мы включили плейер. Если вы воспроизводите файл, то увидите изображение сразу же. Если видна только часть картинки, расширьте сцену. Мы все еще находимся в интерактивном режиме Python, так что поправить сцену можно методом **stage.set_size**.

```
stage.set_size(640,480)
```

Мы также можем поменять состояние плейера, сделав паузу:

```
>>> pipe.set_state(gst.STATE_PAUSED)
<enum GST_STATE_CHANGE_SUCCESS of type
GstStateChangeReturn>
>>> pipe.set_state(gst.STATE_PLAYING)
<enum GST_STATE_CHANGE_ASYNC of type
GstStateChangeReturn>
```

Заметьте, что поток видео действительно приостановился. Вы воображаете, что такое бывает только с файлом, но и с подачей от web-камеры будет то же самое.

Пошалим

Мы всего-навсего создали *Clutter* 'скую версию поточного проигрывателя – как *Kaffeine*, но с меньшим числом опций. Однако тек- »

стура нашего видео – это актер *Clutter*'а, так что с ней можно проделывать всякие штуки.

```
>>> vid.set_size(100,90)
>>> vid.set_size(100,290)
>>> vid.set_size(320,50)
```

Даже когда мы поменяли его размер, или подвинули этого актера...

```
>>> vid.move_by(10,10)
>>> vid.move_by(10,10)
```

К видео можно применять почти все методы актеров. В прошлом выпуске мы детально рассматривали методы анимации, и они сработают и для видеотекстуры. Кратко вспомним, что этот метод берет значение для эффекта анимации (константа *Clutter*, нумерующая различные стили), длительность в миллисекундах и список пар ключ/значение для свойств. Актер преобразует их в абсолютные величины. Например:







› **Минуту назад мы созерцали Пикадилли, а теперь вы уже на Трафальгарской площади, или в Токио, или где-нибудь еще — был бы источник.**

```
>>> vid.animate(clutter.LINEAR, 1000, 'width',640, 'height', 480, 'x',0,'y',0)
<clutter.Animation object at 0x96f3884 (ClutterAnimation at 0xb4799a00)>
>>> vid.animate(clutter.LINEAR, 1000, 'width',320, 'height', 240, 'x',160,'y',120)
<clutter.Animation object at 0x96f4284 (ClutterAnimation at 0xb4799c00)>
```

Чтобы поглубже вникнуть в анимационные преобразования, загляните в прошлый номер. Теперь для нашего маленького эксперимента с камерой у нас есть забава типа картинка-в-картинку; но необходимо разобраться еще с одним элементом. Во-первых, создадим новый видеообъект, такой же, как первый, но с другим потоком:

```
>>> vid2=cluttergst.VideoTexture()
>>> playbin2=vid2.get_playbin()
>>> playbin2.set_property('uri','mmsh://live.camstreams.com/cscamglobal5MSWMExt=.asf')
>>> pipe2=gst.Pipeline("pipe2")
>>> pipe2.add(playbin2)
>>> stage.add(vid2)
>>> pipe2.set_state(gst.STATE_PLAYING)
>>> vid.set_position(0,0)
>>> vid.set_size(100,80)
>>> vid.set_depth(2)
```

С этим мы уже сталкивались ранее, а вот последняя строка появилась впервые. Концепция глубины (depth) – одно из 2.5 измерений *Clutter*. Объекты становятся ближе или дальше от сцены. По некой извращенной (по крайней мере, для меня) логике, для удаления от сцены глубину нужно брать положительной, а для приближения – отрицательной. Итак, если вам нужно наслоить объекты, как мы сделали здесь, возьмите глубину верхнего объекта большей, чем глубина нижнего.

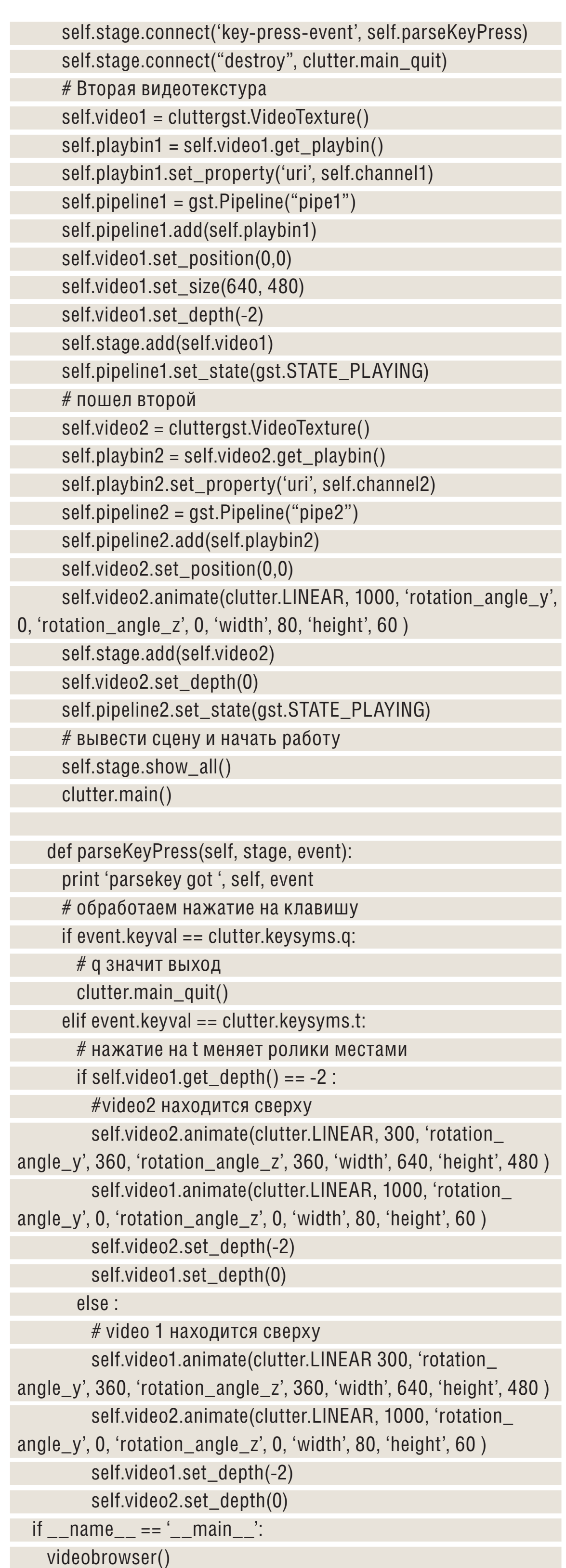
Это меняет способ визуализации актеров, согласно настройкам перспективы, но для малых значений глубины вы ничего и не заметите. Впрочем, поиграйте с параметрами.

Теперь у нас есть все, чтобы создать многопотоковый браузер видео. В следующем куске кода будет создано настоящее приложение Python, так что запустите свой любимый текстовый редактор и введите следующее (или же найдите этот текст на **LXF**DVD):

```
import clutter
import gst
import cluttergst
class videobrowser:
    def __init__ (self):
        self.channel1="mmsh://live.camstreams.com/cscamglobal16?MSWMExt=.asf"
        self.channel2="mmsh://live.camstreams.com/cscamglobal5?MSWMExt=.asf"
        # Инициализируем сцену
        self.stage = clutter.Stage()
        self.stage.set_color(clutter.Color(255, 255, 255, 255))
        self.stage.set_size(640, 480)
        self.stage.set_title('LXF Traffic Watch - press "t" to toggle view')
```

Слово на ушко

Объекты *GStreamer* воспроизводят звук любого видеоресурса автоматически, и если вы просматриваете два источника одновременно, то услышите только какофонию. Для изменения звукового вывода (и, при желании, его перенаправления) пригодятся функции **gst**, а для простой регулировки громкости у актера **VideoTexture** из *Clutter* есть подходящий метод **set_audio_volume()**.



```
        self.stage.connect('key-press-event', self.parseKeyPress)
        self.stage.connect("destroy", clutter.main_quit)
        # Вторая видеотекстура
        self.video1 = cluttergst.VideoTexture()
        self.playbin1 = self.video1.get_playbin()
        self.playbin1.set_property('uri', self.channel1)
        self.pipeline1 = gst.Pipeline("pipe1")
        self.pipeline1.add(self.playbin1)
        self.video1.set_position(0,0)
        self.video1.set_size(640, 480)
        self.video1.set_depth(-2)
        self.stage.add(self.video1)
        self.pipeline1.set_state(gst.STATE_PLAYING)
        # пошел второй
        self.video2 = cluttergst.VideoTexture()
        self.playbin2 = self.video2.get_playbin()
        self.playbin2.set_property('uri', self.channel2)
        self.pipeline2 = gst.Pipeline("pipe2")
        self.pipeline2.add(self.playbin2)
        self.video2.set_position(0,0)
        self.video2.animate(clutter.LINEAR, 1000, 'rotation_angle_y',
 0, 'rotation_angle_z', 0, 'width', 80, 'height', 60 )
        self.stage.add(self.video2)
        self.video2.set_depth(0)
        self.pipeline2.set_state(gst.STATE_PLAYING)
        # вывести сцену и начать работу
        self.stage.show_all()
        clutter.main()

    def parseKeyPress(self, stage, event):
        print 'parsekey got ', self, event
        # обработаем нажатие на клавишу
        if event.keyval == clutter.keysyms.q:
            # q значит выход
            clutter.main_quit()
        elif event.keyval == clutter.keysyms.t:
            # нажатие на t меняет ролики местами
            if self.video1.get_depth() == -2 :
                #video2 находится сверху
                self.video2.animate(clutter.LINEAR, 300, 'rotation_
 angle_y', 360, 'rotation_angle_z', 360, 'width', 640, 'height', 480 )
                self.video1.animate(clutter.LINEAR, 1000, 'rotation_
 angle_y', 0, 'rotation_angle_z', 0, 'width', 80, 'height', 60 )
                self.video2.set_depth(-2)
                self.video1.set_depth(0)
            else :
                # video 1 находится сверху
                self.video1.animate(clutter.LINEAR 300, 'rotation_
 angle_y', 360, 'rotation_angle_z', 360, 'width', 640, 'height', 480 )
                self.video2.animate(clutter.LINEAR, 1000, 'rotation_
 angle_y', 0, 'rotation_angle_z', 0, 'width', 80, 'height', 60 )
                self.video1.set_depth(-2)
                self.video2.set_depth(0)
  if __name__ == '__main__':
    videobrowser()
```

Код **init** создает два видеопотока, размещая один как актера над тем, что больше, а другой помещая в угол. Через несколько секунд потоки оживут, и вы увидите двухэтажные автобусы и прочее. Хитрая часть кода – перехват нажатий клавиш. Если фокус находится в окне, любое нажатие клавиши вызовет событие. Метод **stage.connect()** связывает этот сигнал с методом **parseKeyPress()**, определенным в главном классе приложения. Нажатие клавиши t поменяет два видео местами. Нужно помнить о том, что глубины следует задать заново, чтобы новая, маленькая картинка переместилась вверх (по значению параметра depth определяется, какой актер находился сверху при нажатии кнопки).

Не бойтесь экспериментировать, добавлять экраны и играть с вызовами анимации. *Clutter* выполнит ее за вас, и о действиях после остановки волноваться нет нужды. Клавиша Q привязана к чистому выходу из приложения.



› **Если вам нужны хитрые трюки с каналами и потоками, поможет сайт *GStreamer*.**

Идем дальше

Можно добавлять другие потоки и располагать их каскадом один над другим. Вместо переключения двух потоков, их можно вращать вокруг одной точки. Также можно поиграть с анимацией: значения, задаваемые в методе animate, абсолютны, а не относительны, поэтому угловые повороты кратны 360 градусам – чтобы видеотекстура поворачивалась правильно.

В прошлых двух частях нашего сериала мы рассматривали актеров **clutter.Text()**. Добавить текстового актера, чтобы вы могли видеть, какой поток воспроизводится, или поменять текст при смене экрана, опять-таки не заботясь об анимации, довольно просто. Для потоков аудио, думаю, вам захочется приглушить звук при сворачивании окна. Подробности см. во врезке на стр. 78.

Еще больше штучек с объектами *GStreamer* можно отыскать в документации *GStreamer*: на данный момент она обширнее документации по *PyClutter*. Или взгляните на учебник *GStreamer* в LXF125. LXF

Скорая помощь

К некоторому недоумению, метод **get_uri()** у актера **VideoTexture** ничего не возвращает – ведь это просто текстура; а вот в объекте playbin содержатся данные по URI, и если вы забыли, с чем связан поток, воспользуйтесь **playbin.get_property('uri')**.

Ах, этот Flash!

Некогда любой вид живого видеопотока был простым потоковым сокетом: присосись да пользуйся. Сейчас, кажется, они норовят требовать эти дурацкие плейеры, основанные на Flash. Уж не знаю, зачем: то ли для статистики пользователей, то ли для «защиты» контента… однако наряду с усложнением открытия в *Firefox* нескольких потоков одновременно, это также затрудняет поиск подходящих живых потоков. Хорошая добыча – камеры, отслеживающие дорожное движение; а если вам хочется пошалить, знайте, что некоторые встроенные Flash-плейеры любезно показывают URL сырого потока в исходном тексте страницы. Но я вам этого не говорил. Т-с-с!

» **Через месяц** Мы изучим взаимодействие с актерами *Clutter* и с *Cairo*.

Android: И так

Во второй части учебника по программированию для платформы Android **Джульетта Кемп** объяснит, как улучшить приложение и опубликовать его.



Месяц назад мы настроили среду разработки Android и написали основной код программы для составления списка. Сделаем программу немного более полезной, добавив возможность указать текст записи и создав пункт меню для удаления списка. Потом мы посмотрим, как поделиться этой программой с другими.

Важная особенность архитектуры Android – модульность приложений, и в одной программе можно использовать участки другой, если та это разрешает. А значит, незачем каждый раз разрабатывать собственные компоненты: гораздо эффективнее пустить в ход готовые. Поэтому в приложениях для Android нет единой точки входа – например, в отличие от других программ на Java здесь нет метода **main()**. Вместо этого программы строятся из компонентов. У вашей программы будет стартовая точка по умолчанию (указанная в файле **AndroidManifest.xml**), но она может выполняться и начиная с других точек, в зависимости от того, откуда вызвана. Структура программы – занятия, намерения и представления.

Android – удивительная платформа с быстро растущим сообществом разработчиков. Начать разрабатывать для нее нетрудно, потому что программы почти полностью пишутся на Java. Освоив это, вы сможете заняться массой интересных и новых вещей.



Наш эксперт

Джульетта Кемп очень повеселилась, программируя для Android, и теперь у нее есть список проектов для реализации, за которые она почти наверняка так и не возьмется.

Занятия, Намерения и Представления

Существует четыре основных типа компонентов, но в этой программе используется только первый из них – Занятия:

» **Занятия (Activity)** Каждое Занятие – визуальный интерфейс для определенного действия пользователя в приложении. У нас есть Занятие *List* – отображение списка элементов, и другое Занятие (мы создадим его сегодня), добавление элемента. Занятий может быть столько, сколько необходимо вашему приложению. У Занятия есть окно по умолчанию, полноэкранное либо всплывающее, а содержимое этого окна поставляется Представлениями. Каждое Представление управляет некоторым прямоугольником в окне и наследуется от базового класса **View**. Представления могут откликаться на действия пользователя в «подведомственной» области.

» **Сервисы (Service)** У сервиса нет визуального интерфейса, но он работает в фоне (например, воспроизводит музыку). Обычно он продолжает работать после того, как пользователь покинул приложение.

» **Широковещательные приемники (Broadcast Receiver)** Они принимают и реагируют на широковещательные оповещения, обычно от самой системы. В ответ на некоторые системные оповещения широковещательный приемник может выполнить действие.

» **Контент-провайдеры (Content Provider)** Предоставляют данные приложения в доступ другим программам. Здесь нам это не понадобится, но наше приложение можно будет расширить, сделав его базу данных доступной другим приложениям.



› **Воспользуемся приложением из предыдущей статьи и удалим элементы из созданного списка.**

» **Месяц назад** Мы влезли во внутренности разработки Android.

Далее



› Благодаря открытости исходного кода Android программу в мир может выпустить каждый.

Первые три типа компонентов активируются Намерениями [**Intent**], которые представляют собой объекты, содержащие сообщение. Сообщение может состоять из запроса действия и некоторых данных – мы воспользуемся ими позже на данном уроке.

Удаление элемента – более простое из двух изменений, которые мы собираемся внести в код примера с прошлого урока. Мы воспользуемся контекстным меню: оно появляется, если щелкнуть по элементу и придержать кнопку мыши. В методе **onCreate** уже есть вызов, регистрирующий контекстное меню:

```
registerForContextMenu(getListView());
```

Теперь создадим контекстное меню – точно так же, как создавали обычное. Добавьте следующий метод:

```
public void onCreateContextMenu(Menu menu, View view,
ContextMenuInfo info) {
super.onCreateContextMenu(menu, view, info);
menu.add(0, DELETE_ID, 0, R.string.menu_delete);
}
```

DELETE_ID нужно определить в начале:

```
private static final int DELETE_ID = ADD_ID + 1;
```

Несмотря на то, что элементы добавляются в различные меню, вам понадобится использовать разные значения для двух констант, иначе элемент меню не отобразится должным образом.

Вам также понадобится задать строку **menu_delete** в **res/values/strings.xml**. Как мы обсуждали на прошлом уроке, наборы строк и другие данные приложения лучше хранить в разделе **resources**, а не смешивать с кодом.

```
<string name="menu_delete">Delete item</string>
```

Теперь напишем метод, обрабатывающий события при выборе пункта меню. Создайте метод из файла **snippet1.txt** в каталоге **Magazine/Android** нашего DVD. Опять же, в выражении **switch** есть только одна ветвь. Обратите внимание на разницу имен этого метода и метода **onMenuItemSelected()**, который мы написали в предыдущей статье. Эти методы предоставляет Android: сразу после создания меню методы с такими именами будут автоматически использованы для обработки его пунктов. Больше ничего делать не нужно.

Так как это контекстное меню, оно автоматически связывается с определенным элементом списка. Метод получает информацию о выбранном элементе, вызывает обработчик базы данных для удаления записи и снова получает данные из базы, так что изменение немедленно отображается на экране. Наконец, нам потребуется метод удаления данных в базе в **ListDbAdapter.java**:

```
public boolean deleteItem(long rowId) {
return mDb.delete(DB_TABLE, DB_ROWID + "=" +
rowId, null) > 0;
}
```

Он использует переданный идентификатор строки **rowId** и удаляет указанный элемент. Данный идентификатор передается из метода **onContextItemSelected()**, в котором он извлекается из базы данных.

Если вы не используете *Eclipse*, придется вручную импортировать классы в начале файла **List.java** – см. код на DVD.

Добавляем элемент с заголовком

До настоящего времени все новые элементы списка получали один и тот же стандартный текст, но на практике толку от этого мало. Сделаем так, чтобы пользователь мог ввести то, что сочтет нужным.

Для этого создадим еще одно Занятие – 'ListAdd'. Это будет класс **ListAdd**, который будет жить в файле **ListAdd.java**. Как говорилось выше, Занятие – нечто выполняемое внутри приложения, и у него обычно есть собственные установки экрана, оно обрабатывает собственные Представления и т.д.

Прежде всего нам понадобится код, вызывающий **ListAdd** из главного Занятия **List**. Измените метод **createItem()** в **List.java** таким образом:

```
private void createItem() {
Intent i = new Intent(this, ListAdd.class);
startActivityForResult(i, ITEM_ADD);
}
```

Намерение — способ реализации обмена информацией между Занятиями. В него мы передаем текущий контекст [**this**] и имя класса, которому мы собираемся передать объект **Intent**. Затем мы вызываем класс методом **startActivityForResult()**, передавая ему полученный объект Намерения и код, которым он воспользуется для идентификации результата. Вам также потребуется установить его значение в начале файла – подойдет любое:

```
private static final int ITEM_ADD=0;
```

Вместо **startActivityForResult()** вы можете воспользоваться **startActivity()**, но его имя подсказывает, что **startActivity()** не возвращает ничего из Занятия, которое он запускает. Здесь это не подходит, так как мы хотим получить строку и обработать ее. В общем, метод **startActivityForResult()** более удобен.

Далее обработаем строку, возвращаемую из действия **ListAdd**. Добавьте метод **onActivityResult()** – он выглядит следующим образом:

```
@Override protected void onActivityResult(int requestCode,
int resultCode, Intent i) {
super.onActivityResult(requestCode, resultCode, i);
if (resultCode == RESULT_CANCELED) {
return;
}
Bundle extras = i.getExtras();
switch(requestCode) {
```

»



Чтобы запустить эмулятор, наберите **emulator -avd myavd**. Для переустановки в эмулятор существующего пакета, наберите **adb install -r List/bin/List-debug.apk**.

```
case ITEM_ADD:
String label = extras.getString(ListDbAdapter.DB_ITEM);
mDb.createItem(label);
getData();
break;
}
}
```

Этот метод обрабатывает все результаты всех Занятий, которые вызывает **List**: поэтому мы и передаем код идентификации. Здесь у нас единственное доступное значение, но если бы вы захотели делать что-то еще, например, редактировать элементы списка, потребовалось бы больше.

Сперва проверим, была ли операция отменена – тогда делать ничего не нужно. Значение переменной **RESULT_CANCELED** устанавливается в главном классе Занятия **Activity**, так что это глобальное возвращаемое значение. В классе **ListAdd** мы свяжем этот результат с кнопкой Cancel [Отмена].

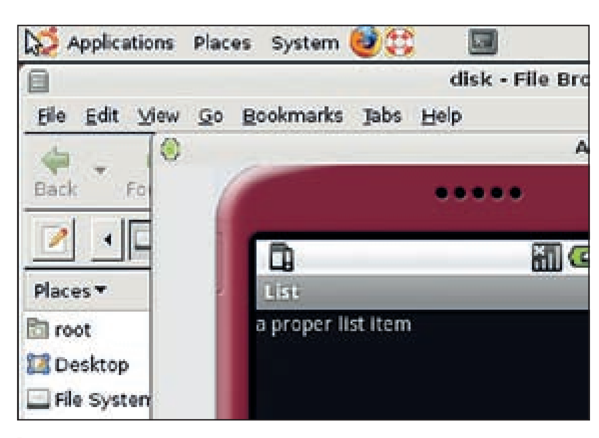
Если Занятие не было отменено, мы берем связку (**Bundle**), ассоциированную с возвращенным Намерением. Она используется для получения карты дополнительных данных (**extras**), которая хранится внутри Намерения (мы заполним ее в классе **ListAdd** позже) и связывает строки с другими типами данных: это нечто вроде хэша, который возвращает данные по ключу. Как видите, в секции **ITEM_ADD** оператора switch мы получаем значение **label** (имя элемента, который будет добавлен в список) из **extras** по ключу **ListDbAdapter.DB_ITEM**. Для ключа понадобится статическая переменная (вы будете обращаться к ней из нескольких классов), и неплохо использовать существующую и релевантную статическую строку. В данном случае, это столбец базы данных, где будут храниться данные. Затем вызывается метод обработчика базы данных, и метод **getData()** обновляет данные на экране.

«Для отладки кода воспользуйтесь ddms и проверьте файлы журналов.»

Последнее, что нужно сделать – создать класс, который сделает всю работу. Взгляните на код класса **ListAdd** на нашем DVD, в нем всего один метод:



› Вот что вы увидите при добавлении элемента в список.



› Эмулятор Android — отличный способ тестировать свои программы перед выпуском прямо на рабочем столе Linux.

```
@Override
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.newitem);
label = (EditText) findViewById(R.id.newitemlabel);
Button okButton = (Button) findViewById(R.id.okbutton);
cancelButton = (Button) findViewById(R.id.cancelbutton);
okButton.setOnClickListener(new View.OnClickListener()
{
public void onClick(View view) {
Bundle bundle = new Bundle();
bundle.putString(ListDbAdapter.DB_ITEM,
label.getText().toString());
Intent i = new Intent();
i.putExtras(bundle);
setResult(RESULT_OK, i);
finish();
}
});
cancelButton.setOnClickListener(new View.
OnClickListener() {
public void onClick(View view) {
Intent i = new Intent();
setResult(RESULT_CANCELED, i);
finish();
}
});
}
```

Этот метод вызывается при активации Занятия из **List** (а вообще-то, и из любого другого Занятия). Устанавливается раскладка Представлений (см. файл **newitem.xml** на DVD – в нем есть блок **EditText**, куда можно ввести текст, и кнопка **OK**).

Мы получаем значение **label** (представляющее собой объект **EditText**, объявленный в начале класса) из раскладки по его идентификатору ID **newitemlabel**, а затем также получаем кнопку **OK** из раскладки как объект, снова идентифицируя его по ID. То же происходит и для кнопки **Cancel**.

Самая важная часть кода – метод **setOnClickListener()** для кнопки **OK**: он определяет, что произойдет по ее нажатию. Создается новая связка [Bundle], и к ней добавляется пара: ключ (строка **DB_ITEM**, используемая в коде обработки возврата в классе **List**) и значение из блока **EditText**, преобразованное в строку. Наконец, создается Намерение, принимающее связку; результат устанавливается в **OK** (**RESULT_OK** – стандартное значение, заданное в классе **Activity**), и Занятие завершается. Затем результат вернется в вызывающее Занятие (в нашем случае – **List**), которое обработает его в соответствии с уже написанным нами кодом.

Нам также понадобится метод **setOnClickListener()** для кнопки Cancel. Он проще: в нем лишь создается пустой объект Intent и используется глобальная константа **RESULT_CANCELED**, означающая, что действие было отменено. Последняя задача – добавить свой класс в файл **AndroidManifest.xml**, иначе он не будет найден во время компиляции. Добавьте строку

```
<activity android:name=".ListAdd"></activity>
```

после тэга **</activity>**, завершающего листинг действия **.List**.

Скомпилируйте свой код (**ant debug**), запустите эмулятор, загрузите его и попробуйте! Для отладки кода воспользуйтесь **ddms** и проверьте файлы журналов, детали работы с которыми описаны в предыдущей статье.

Публикация приложения

Пока у вас получилась отладочная (debug) версия программы. Следующий шаг – сделать из нее окончательную (release) версию, которой можно будет поделиться с другими. Все приложения Android должны быть заверены сертификатом разработчика, который может быть самоподписанным. Также нужно задать переменные **android.versionCode** и **android.versionName** в разделе **<manifest>**.

Перед публикацией вам следует протестировать вашу программу на настоящем устройстве, если это вообще возможно. Не мешает также нарисовать иконку и указать ее в атрибуте **android.icon** секции **<application>** файла **AndroidManifest.xml**. Сам файл иконки сохраните в каталоге **res/drawable**.

Редактируя **AndroidManifest.xml**, нужно также указать в нем значение **android.label**, название вашей программы в меню приложений. Удалите из манифеста атрибут **android:debuggable=«true»** и убедитесь, что в коде программы не осталось отладочных команд.

Первый шаг – создание окончательной версии, пока не подписанной. Это сделает команда **ant release** (в отличие от **ant debug**). Далее создайте свой сертификат с помощью *Keytool* (проверьте, что ссылка из **/usr/bin/keytool** указывает на версию в JDK, а не **gjc**). Рекомендуется задать срок действия в 25 лет, чтобы гарантировать возможность будущих обновлений без проблем для пользователя (сертификаты будут проверяться во время установки/обновления). Если вы хотите разместить приложение на Android Market, вам придется указывать именно этот срок. Он соответствует 9130 дням, которые мы округлим до 10 000.

```
keytool -genkey -v -keystore ~/android/release-keys.keystore
-alias listapp -keyalg RSA -validity 10000
```

У вас запросят пароль для **keystore**, кое-какую персональную информацию о вас и пароль для ключа. Используйте сложные пароли или парольные фразы. Для **keystore** можно указать любой путь.

Затем подпишите приложение с помощью *Jarsigner*, запустив эту команду в подкаталоге **/bin/** директории вашей программы:

```
jarsigner -verbose -keystore ~/android/release-keys.keystore List-
unsigned.apk listapp
```

Здесь **listapp** – псевдоним для ключа, **List-unsigned.apk** — само приложение, а **-keystore** задает ключевое хранилище – нужно указать полный путь к нему.

Итак, приложение подписано; дайте ему какое-нибудь значимое имя, используя **mv List-unsigned.apk List.apk**. Теперь можно проверить, что программа подписана, командой:

```
jarsigner -verify List.apk
```

Еще раз проверьте финальную версию, и затем ее можно публиковать. Самый простой способ сделать это – просто выложить файл **.apk** на ваш сайт: пользователи смогут загрузить его и установить приложение с помощью *AppManager* или похожей программы.

Альтернатива – воспользоваться Android Market (http://market.android.com/publish), который требует регистрации и взноса в 25 долларов, но упрощает загрузку приложения.



› На сайт Android Market можно зайти прямо с телефона Android.

Будем надеяться, наши уроки дали вам общее понятие о написании программ для Android. В сети есть масса полезных ресурсов: очень хороша документация разработчика для Android, есть и форумы, где можно задать вопрос разработчикам. Чтобы упростить создание интерфейса, попробуйте *DroidDraw* (www.droiddraw.org) – созданный нами интерфейс довольно примитивен! Также можно взглянуть на расширение Google API: оно позволит связать вашу программу с Google Maps. Как мы уже говорили, архитектура Android позволяет вызывать части кода других программ: приложение со списком Намерений (Intents List Appendix) в документации разработчика Android содержит все программы Google, которые можно вызвать (например, браузер или программу для звонков по телефону), и сообщает как это сделать. Вы также можете рассмотреть API для определения движения. LXF

Дальнейшие шаги

Наши уроки вас заинтриговали? Не останавливайтесь на достигнутом: есть масса способов улучшить написанное нами приложение. Вот несколько из них для начала. Обратите внимание, что некоторые из них могут потребовать изменения – или полной переработки – кода для работы с базой данных.

- » Добавить возможность редактирования элемента: можно воспользоваться классом **ListAdd** и переданным объектом **Intent** для хранения информации о текущих значениях метки элемента и идентификатора строки в базе данных.
- » Добавить обработку ввода, чтобы, например, нажатие Enter обладало тем же эффектом, что и нажатие кнопки OK в окне «Добавить элемент».
- » Улучшить управление жизненным циклом, чтобы кнопка «Назад» работала правильно.
- » Добавить флажки рядом с каждым элементом.
- » Реализовать массовое удаление.
- » Добавить к элементу категорию, поле примечания или другие информационные поля.
- » Поищите сайты программ для составления списков и подумайте, не поучаствовать ли.

OpenVZ: Ас вир-

Избегаете виртуализации из-за нагрузки на процессор? **Маянк Шарма** покажет невероятно быстрый способ, даже без специального оборудования.





Наш эксперт

Маянк Шарма написал две книги по администрированию *Elgg* и *Openfire* и последние четыре года был пишущим редактором на Linux.com.

Программное обеспечение для виртуализации бывает всевозможных форм и размеров. То, что предназначено для пользователей настольных систем, такое как *VMware Workstation* и Sun *VirtualBox*, снабжено легким в использовании графическим интерфейсом, но за это приходится платить производительностью. Более продвинутое, типа *KVM* и *Xen*, быстрее, но требует специальных процессоров с расширением для виртуализации. Ну, а еще есть *OpenVZ*.

Свободное от компьютерного жаргона определение *OpenVZ* дать достаточно сложно. С его помощью вы запускаете множество установок операционных систем внутри множества изолированных областей, известных как контейнеры. Это подобно механизму Jails из FreeBSD и технологии Solaris Zones.

Контейнеры делают *OpenVZ* идеальным для запуска служб, подверженных атакам извне – web- или FTP-серверов: если кто-то и взломает их, добычей станет только контейнер данного сервера. Поэтому в контейнерах обычно запускается минимальная установка ОС, позволяющая поддерживать только определенный сервис и ничего больше.

Главное отличие *OpenVZ* от других решений виртуализации заключается в скорости. *OpenVZ* работает на системном уровне, что делает его весьма энергичным – на загрузку уходит секунд пять, а приложения запускаются вообще мгновенно. Расплатой за скорость является невозможность запускать внутри контейнеров ничего, кроме Linux. Мы не можем помочь вам найти причину для запуска Linux внутри Linux, но вот подскажем, как заставить его работать с *OpenVZ*.

Прежде чем начать, прочитайте врезку про требования к аппаратным узлам (это то, что в *OpenVZ* зовется хостом), где говорится о настройке хоста.

Так как *OpenVZ* работает на уровне операционной системы [его обычно называют уровнем ядра, – *прим. пер.*], то он нуждается в слиянии с ядром. Мы сделаем это всего одной командой. Заберите двоичный RPM-пакет ядра *OpenVZ* с http://wiki.openvz.org/Download/kernel. Там имеются три ветви, различающиеся по стабильности. Сайт рекомендует ветвь RHEL5. Каждая ветвь содержит перечень ядер; выбрать нужно то, что подходит к вашему оборудованию. Там есть различные ядра для мультипроцессорных хостов (узлов) и для машин с более чем 4 ГБ ОЗУ. Все ядра доступны в версиях и для 32-битных, и для 64-битных процессоров. Чтобы выбрать правильное ядро, тщательно изучите информацию на странице загрузки. Допустим, вам требуется типовое, суперстабильное ядро RHEL4 – установите его командой

```
rpm -ivh ovzkernel-2.6.9-023stab051.3.i686.rpm
```

Когда ядро установится, оно заодно модифицирует ваш *Grub* для загрузки в ядро *OpenVZ*. Но это потребует некоторой доводки, поэтому откройте конфигурационный файл *Grub* (обычно **/boot/grub/grub.conf**) и поищите загружающую ядро запись со словом **stab**.

Давайте его загрузим

Так как эта строка загружает ядро *OpenVZ*, для ясности измените заголовок на *OpenVZ*. Также удалите лишние загрузочные параметры ядра, чтобы строки выглядели примерно так:

```
title OpenVZ (2.6.9-023stab051.3)
    root (hd0,0)
    kernel /vmlinuz- 2.6.9-023stab051.3 ro root=/dev/sda5
```



› ***Iceweasel*, запущенный на аппаратном узле CentOS из контейнера Debian.**

» **Месяц назад** Мы закинули Ubuntu в облака, на сервера EC2 от Amazon.

туализации

```
initrd /initrd-2.6.9-023stab051.3.img
```

Убедитесь, что эта запись выбрана для загрузки по умолчанию.

Следующим шагом нужно поправить файл **/etc/sysctl.conf** для установки некоторых параметров ядра для *OpenVZ*. Для этого нужно всего лишь добавить в файл следующие строки:

```
net.ipv4.ip_forward = 1
net.ipv4.conf.default.proxy_arp = 0
net.ipv4.conf.all.rp_filter = 1
kernel.sysrq = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

OpenVZ не очень дружит с SELinux, поэтому обеспечьте, чтобы у вас в **/etc/sysconfig/selinux** стояло **SELINUX=disabled**.

Наконец, скачайте два инструмента *OpenVZ* – *Vzctl* и *Vzquota* (опять же, оба зависят от вашего оборудования) с http://wiki.openvz.org/Download/utils и установите их с помощью:

```
rpm -ivh vzctl-*.rpm vzquota-*.rpm_
```

Теперь перезагрузите свой аппаратный узел, что должно активировать службу *OpenVZ* и привести все «в движение».

Готовые контейнеры

Контейнер в *OpenVZ* – почти то же, что виртуальная машина в *VirtualBox*. По существу, это полноценная Linux-система. Вы можете либо пройти через процесс создания своего собственного контейнера, либо взять готовый с сайта *OpenVZ* (http://wiki.openvz.org/Download/template/precreated). Существуют и созданные пользователями контейнеры, и все вместе они покрывают основные популярные дистрибутивы. Контейнеры избавлены от таких крайностей, как *X*, и составляют лишь часть от реального размера дистрибутива.

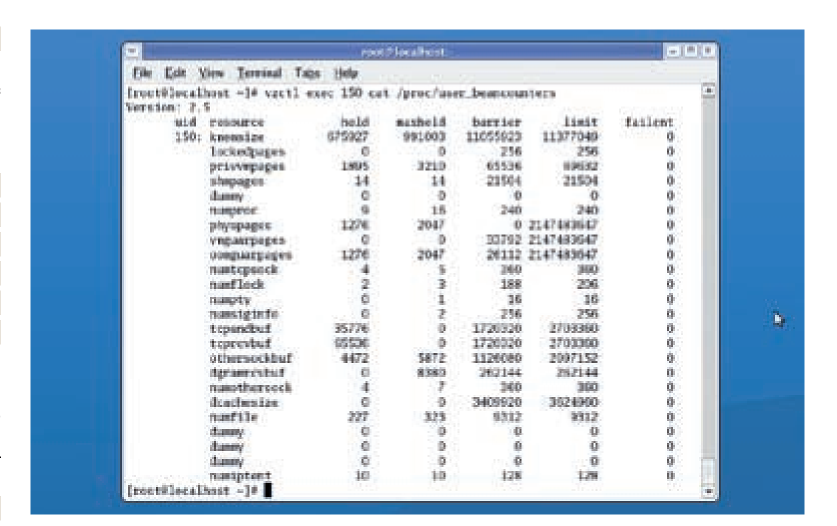
Скачав шаблон, поместите его в директорию **/vz/template/cache** на аппаратном узле. Теперь трансформируем этот контейнер в рабочую Linux-машину, известную как Virtual Private Server [Виртуальный частный сервер] или VPS.

Каждый VPS имеет уникальный ID – *OpenVZ* резервирует ID от 0 до 100. Хорошей идеей будет использовать в качестве ID последнюю цифру IP-адреса, который вы выдадите VPS. Держа это в голове, создадим VPS с 32-битным шаблоном Debian-3.1:

```
# vzctl create 150 --ostemplate debian-3.1-x86 --config vps.
basic
Creating container private area (debian-3.1-x86)
Performing postcreate actions
Container private area was created
```

У всех VPS есть личное пространство и параметры настройки на основе эталонного файла конфигурации. Личное пространство, созданное из шаблона – это директория, содержащая файлы, которые относятся к VPS. Оно монтируется как **/vz/root/<vpsid>/** на аппаратном узле и предоставляет пользователям VPS полноценное дерево файловой системы Linux. Примеры конфигурационных файлов находится в директории **/etc/sysconfig/vz-scripts**. Файл **vps.basic** содержит параметры управления ресурсами, подходящие для большинства VPS. Пробудите VPS с помощью:

```
# vzctl start 150
Starting container ...
Container is mounted
Setting CPU units: 1000
```



› **Список ресурсов поможет отследить их нехватку для запуска приложений.**

```
Configure meminfo: 65536
Container start in progress...
```

Независимо от вашего оборудования, это займет не более пяти секунд – вдумайтесь: за столь короткое время запустится полноценная Linux-система. Видел бы это *VirtualBox*! Давайте исследуем наше новое творение чуть подробнее, набрав

```
#vzlist -a
CTID   NPROC STATUS  IP_ADDR    HOSTNAME
 150       8 running -          -
```

Эта команда показывает список всех VPS на аппаратном узле, их текущее состояние, сетевую информацию и число запущенных процессов. Вы можете проверить последнее с помощью

```
# vzctl exec 150 ps
  PID TTY        TIME CMD
    1 ?      00:00:00 init
20362 ?      00:00:00 syslogd
20371 ?      00:00:00 sshd
20388 ?      00:00:00 xinetd
20468 ?      00:00:00 cron
20473 ?      00:00:00 apache2
21505 ?      00:00:00 sendmail-mta
21510 ?      00:00:00 ps
```

»

Идеи по аппаратным узлам

Как и с остальными программами виртуализации, надежность и отказоустойчивость аппаратуры, на которой запускаются виртуальные машины, являются критическими параметрами. Сбой аппаратного узла перечеркнет все преимущества *OpenVZ*. На производственных машинах хорошей идеей будет использовать для хранения данных VPS RAID-массив.

Хотя *OpenVZ* не имеет никаких специфических требований к аппаратуре и прекрасно себя чувствует на Celeron с 256 МБ ОЗУ, лучше найти машину помощнее, чтобы удовлетворить потребности сервисов в запущенных контейнерах.

Устанавливая дистрибутив Linux на аппаратный узел, убедитесь, что он стабилен и имеет промышленный уровень. Мы использовали CentOS 4 с суперстабильным ядром RHEL 4, но CentOS 5 с ядром RHEL 5 – тоже хорошее решение. Разбивая жесткий диск аппаратного узла при установке, желательно создать отдельный раздел **/vz** для размещения шаблонов и VPS отдельно от основного корня. Наконец, документация *OpenVZ* рекомендует использовать для этого раздела файловую систему ext3.

Ключ **exec** запускает любую команду в заданном VPS. Вы можете также использовать *vzctl* для установки пароля пользователя:

```
# vzctl set 150 --userpasswd root:нечто_длинное
```

Теперь можно войти в VPS через SSH от имени root и администрировать его как обычный самостоятельный Linux-компьютер. В качестве альтернативы можете использовать оболочку, предоставляемую VPS. Для этого зайдите в VPS с помощью

```
# vzctl enter 150
entered into CT 150
localhost:/#
```

А теперь напрягитесь чуть больше и создайте пользователя, как если бы вы были в обычной командной строке Linux:

```
:/# useradd bodhi
:/# passwd bodhi
:/# mkdir /home/bodhi
:/# chown bodhi /home/bodhi
```

Для выхода наберите

```
:/# exit
logout
exited from CT 150
```

Наигравшись, обеспечьте автозапуск VPS после перезагрузки аппаратного узла:

```
# vzctl set 150 --onboot yes --save
```

Параметр **--save** делает настройки постоянными. Не добавляйте его, если вы экспериментируете с настройками. Для удаления VPS, предварительно убедившись, что он не запущен, используйте

```
# vzctl stop 150
Stopping container ...
Container was stopped
Container is unmounted
# vzctl destroy 150
Destroying container private area: /vz/private/150
Container private area was destroyed
```

Конфигурационный файл VPS не уничтожается, но к его имени добавляется **.destroyed**, например: **/etc/sysconfig/vz-scripts/150.conf.destroyed**. Вы можете удалить его вручную.

Скорая помощь

Если VPS жалуется, что ваше ядро устарело, и отказывается стартовать, укажите новую версию ядра в **/proc/sys/kernel/virt_osrelease**.

Скорая помощь

Вы можете модифицировать настройки вашего VPS прямо в ходе его работы.

Сеть и ПО

Теперь перейдем к реальным задачам, таким как сеть и установка приложений. Как любая машина, VPS нуждается в IP-адресе, чтобы быть доступным в сети. Для простоты мы будем использовать Network Address Translation (NAT). Вы можете также установить пакет **bridge-utils** на аппаратный узел и настроить мост. Хорошей новостью насчет *OpenVZ* будет то, что он позволяет назначать и удалять сетевые настройки даже при работающем VPS.

› ***WebVZ* — прекрасный графический интерфейс для новичков, напуганных *Vzctl*.**

Первым делом проверьте, что у вас на аппаратном узле включена трансляция IP-пакетов, набрав

```
# cat /proc/sys/net/ipv4/ip_forward
1
```

Оставаясь на аппаратном узле, включите NAT:

```
# iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth0 -j
SNAT --to 192.168.2.3
```

Чтобы заставить эту команду работать с вашей сетью, вам нужно модифицировать только три вещи. Сперва замените **10.0.0.0/8** на диапазон частных IP-адресов, которые вы хотите дать VPS. Далее, замените **eth0** на ваше сетевое устройство и **192.168.2.3** на IP аппаратного узла. Когда закончите, задайте IP-адрес для вашего VPS с помощью

```
# vzctl set 150 --ipadd 10.1.1.150 --save
Adding IP address(es): 10.1.1.150
Saved parameters for CT 150
```

Затем задайте DNS:

```
# vzctl set 150 --nameserver 208.67.222.222 --save
File resolv.conf was modified
Saved parameters for CT 150
```

Аналогично можно также задать для VPS имя хоста. Проверьте настройки с помощью **ifconfig** внутри VPS и попробуйте пропинговать что-нибудь внешнее, чтобы убедиться в правильности параметров.

Запустив сеть, приступим к загрузке приложений в ваш VPS. Эта процедура не отличается от обычных дистрибутивов Linux – потребуется только настройка менеджера пакетов. Так как наш VPS «крутит» Debian, это будет почтенный *apt-get*. Отредактируйте файл **/etc/apt/sources.list** внутри VPS, как если бы это был Linux на обычной машине, и добавьте или удалите репозитории. Затем извлеките все, что вам нужно, из репозиториев с помощью **apt-get install**.

Запуск графических приложений все же требует больше кунг-фу командной строки, чем простое использование *apt-get*. Давайте попробуем сделать это с *Iceweasel*. Сперва убедимся, что в конфигурационном файле *sshd* (обычно это **/etc/ssh/sshd_config**) включен *X*-проброс внутри VPS:

```
X11Forwarding yes
```

Также нужно обеспечить установку в VPS пакета **xauth**. Если его нет, добавьте его из локального репозитория.

Наконец, перезапустите SSH-сервис:

```
:/# /etc/init.d/ssh restart_
```

Теперь вернитесь назад на аппаратный узел подключитесь к VPS через SSH:

```
# ssh -2 -c blowfish -X bodhi@10.1.1.150
```

Тут вас попросят ввести пароль пользователя. После успешной аутентификации, просто запустите приложение, подивившись на скорость:

```
$ iceweasel
```

Управляем ресурсами

Большая разница между *OpenVZ* и другими известными решениями для виртуализации заключается в том, что VPS могут масштабироваться для потребления всех ресурсов аппаратного узла. Чтобы ограничить это, можно задать жесткие лимиты на использование ресурсов – дискового пространства, памяти или нагрузки на процессор. Существует несколько способов изменения настроек контейнера. Один использует команду **vzctl**, второй заключается в правке конфигурационного файла. Для нашего Debian-контейнера им будет **/etc/sysconfig/vz-scripts/150.conf**.

Давайте ограничим использование процессора нашим контейнером. *OpenVZ* учитывает потребление CPU в процентах от полной мощности аппаратного узла. Эта мощность рассчитывается на основе специального алгоритма *OpenVZ*. Вы можете проверить величину полной мощности вашего аппаратного узла, выполнив:

```
#vzcpucheck
```



```
Current CPU utilization: 2000
Power of the node: 103730
```

Чтобы выделить 10 % процессора контейнеру, сперва вычислим значение мощности (103,730 * 0.1) – округленно, это даст нам 1500 единиц. Также нам необходимо задать максимальное превышение, чтобы убедиться, что контейнер не вызовет перегрузки процессора. Ограничим контейнер 20 % от полного использования, набрав следующее:

```
# vzctl set 150 --cpuunits 1500 --cpulimit 20 --save
Setting CPU limit: 20
Setting CPU units: 1500
Saved parameters for CT 150
```

Другая существенная проблема с виртуальными машинами заключается в том, что при израсходовании места на диске выделить дополнительное бывает проблематично. Но не в случае *OpenVZ*! Прежде чем добавить виртуальной машине места, проверьте текущее состояние с помощью

```
# vzctl exec 150 df -h
Filesystem  Size  Used  Avail  Use%  Mounted on
simfs  1.0G  625M  400M  62%  /
tmpfs  187M  0  187M  0%  /dev/shm
```

В зависимости от применения вашего VPS, изначальных 1 ГБ может не хватить. Прежде чем увеличить его до 5 ГБ, уясним две вещи. Первая – как и настройки CPU, дисковое пространство сразу целиком не выделяется. Вместо этого задается предел, до которого может расти контейнер. Вторая – необходимо задать максимальный лимит (например, 6 ГБ), что даст вам буфер в 1 ГБ. Такое свободное пространство позволяет добавить VPS места без фатальных ошибок диска. Итак, чтобы увеличить диск до 5 ГБ с максимальным лимитом в 6 ГБ, используйте

```
# vzctl set 150 --diskspace 5G:6G --save
Saved parameters for CT 150
```

Существуют и другие вещи, на которые можно повлиять. Взгляните на конфигурационный файл контейнера, а также на man-страницу *Vzctl*, чтобы найти параметры, с которыми вы можете поиграть.

Графический интерфейс

Если вы любитель пощелкать мышью, не отворачивайтесь от нашего урока из-за всех этих командных строк: у *OpenVZ* есть графический web-интерфейс, называемый *WebVZ*. С его помощью можно создавать контейнеры, перемещать их с одного аппаратного узла на другой, модифицировать настройки, конфигурировать сеть, следить за ресурсами и делать много чего еще.

WebVZ требует Ruby on Rails, поэтому вам нужно разжиться **ruby** и **rubygem** через менеджер пакетов или установив их из исходных кодов. *Gems* – это менеджер пакетов Ruby, и вы можете привлечь его для установки Rails простой командой **gems install rails**. Помните, что это надо делать на аппаратном узле, а не внутри VPS.

WebVZ также требует базу данных *SQLite* и ее библиотеки разработчика. Обратитесь к менеджеру пакетов аппаратного узла и установите пакет **libsqlite3-dev**, который также добавит менеджер базы данных. Для подключения его к Ruby, используйте менеджер пакетов *Gem* для установки коннектора *SQLite*:

```
gem install sqlite3-ruby
```

Когда все будет подготовлено, заберите последний tar-архив *WebVZ* с http://bit.ly/c9YFPJ и распакуйте его в какую-либо директорию. Запустите сервер *WEBrick*, управляющий страницами *WebVZ*:

```
$ cd webvz
$ ruby script/server
=> Booting WEBrick
=> Rails 2.3.4 application starting on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2009-11-18 20:00:19] INFO  WEBrick 1.3.1
[2009-11-18 20:00:19] INFO  ruby 1.8.7 (2009-06-12) [i686-linux]
[2009-11-18 20:00:19] INFO  WEBrick::HTTPServer#start: pid=23235 port=3000
```

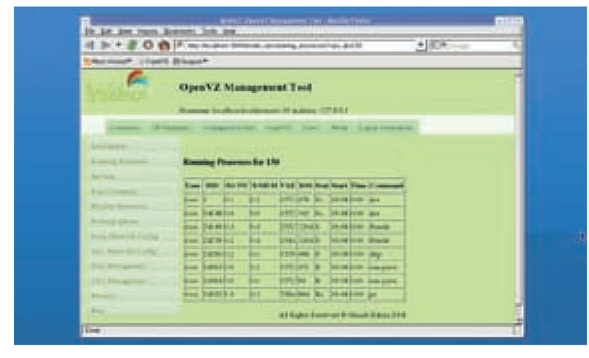
Вывод сообщает, что сервер запущен на **localhost** и доступен на порту **3000**. Открыв браузер, укажите ему http://localhost:3000. Имя пользователя и пароль по умолчанию – **admin** и **admin123**.

Все, кто содержат собственные сайты или запускают сервер файлообмена между домом и работой, должны пересесть на *OpenVZ* – это намного быстрее и безопаснее. Вот вам и достаточное основание для запуска Linux внутри Linux. **LXF**

› **LiveCD *OpenVZ* CentOS имеет больше параметров ядра, чем Knoppix.**



Вики *OpenVZ* содержит подробную информацию о продвинутых задачах, таких как перенос физической установки Linux в контейнер.



› **Показан *OpenVZ* с тремя запущенными виртуальными машинами на 1,3-ГГц Celeron с 256 МБ ОЗУ. И старт почти мгновенный!**

Live CD OpenVZ

Хотите избежать утомительного процесса подготовки аппаратного узла и оснащения его подходящим ядром? Тогда раздобудьте LiveCD *OpenVZ*. Это идеал для оценки возможностей платформы виртуализации *OpenVZ*. Есть два вида LiveCD – один базируется на Knoppix 5.1.1, а другой на CentOS 4.4. Вы можете скачать их с http://wiki.openvz.org/Download_live_CD.

LiveCD предлагают различные ядра *OpenVZ*, включая серию 2.9 на основе RHEL 4, серию ядра 2.18 на основе RHEL 5, новое ядро 2.20 и т. п. Также присутствуют три готовых контейнера для создания минимальных VPS на основе 32-битной версии CentOS 4, Fedora Core 5 и Debian 3.1. Они заодно укомплектованы *Vzctl* и *Vzquota*.

С помощью LiveCD вы можете полностью исследовать *OpenVZ* – от создания контейнеров до управления их ресурсами. А вот чего вы не сможете сделать – это установить LiveCD на жесткий диск; вдобавок на нем катастрофически мало места для развертывания GUI *WebVZ*.

» **Через месяц** Защититесь от сбоев диска программной заменой RAID.





Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru

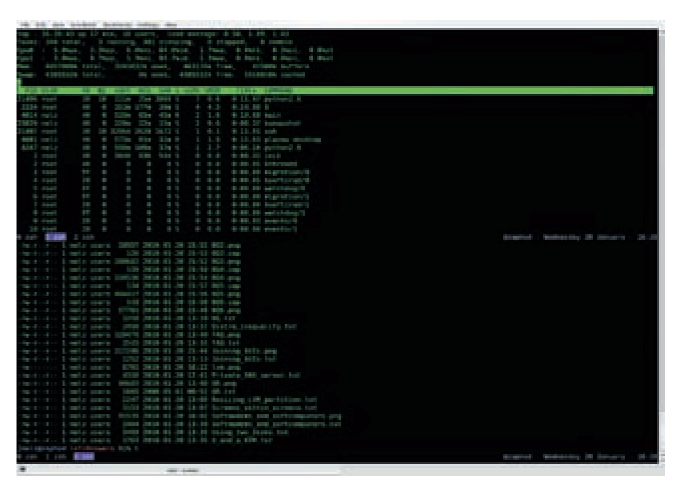
В этом месяце мы ответим на вопросы про:

1 Экраны в экранах

В Иногда мне требуется пользоваться командой runlevel 3. Поскольку я делаю это из *Konsole*, мне, к сожалению, приходится открывать множество терминальных сеансов. Не подскажете ли вы какой-нибудь более изящный способ?

Мохд Ридзуан Бистами [Mohd Ridzuan Bistami]

О Подходящая Вам программа называется *Screen*. Это стандартный инструмент GNU, и он должен иметься в репозиториях Вашего дистрибутива. *Screen* можно назвать оконным менеджером для терминалов или виртуальным терминалом. Запустите эту программу из терминала; помимо них, она работает и в виртуальной консоли или X-терминале, и на вид ничего не произойдет. Но на самом деле Ваш терминальный сеанс заменится сеансом *Screen*; он выглядит так же, как терминальный, и разницы Вы не ощутите, пока не освоите магические трюки этой программы. Запустите команду вывода какой-нибудь информации – допустим, **ls** – и нажмите комбинацию клавиш Ctrl+A, затем C. На первый взгляд Вам покажется, что терминальный вывод исчез, однако это заблуждение: Вы просто создали новое окно *Screen*. Запустите какую-нибудь команду и здесь – скажем, top – и нажмите клавиши Ctrl+A, а затем – Shift+N. Вывод команды top исчезнет, и вместо него появится вывод команды ls. Чтобы переключиться обратно, снова нажмите клавиши Ctrl+A N.



Программа *Screen* управляет несколькими сеансами терминала в одном окне или консоли.

При запуске, *Screen* открывает единственное окно – в нем вы давали команду **ls**. Комбинация Ctrl+A – командная последовательность *Screen*: клавиша, нажатая вслед за ней, воспринимается как команда. Клавишей C создается новое окно, клавиша N переключает на следующее окно, а цифры от 0 до 9 перенесут Вас в окно с соответствующим номером. Итак, Вы создали новое окно, запустили в нем команду, а потом переключились между двумя окнами. Закрывается окно *Screen* точно так же, как обычный сеанс терминала: командой **logout** или нажатием комбинации Ctrl+D. После закрытия последнего окна завершится и сама программа.

Первое, что я делаю в *Screen* сразу же после его установки – добавляю следующие строки в конец файла **/etc/screenrc**:

```
# Заголовок в последней строке
caption always "%{= kw}%-w%{= BW}%n
%t%{-}%+w %-= @%H -%LD %d %LM -%c"
```

Эти команды добавляют в нижнюю часть всех окон *Screen* строку состояния, отображающую список окон и имя хоста (что полезно при запуске *Screen* через SSH). Опции можно помещать в файл **/etc/screenrc** или в пользовательский файл **.screenrc**. Помимо прочего, Вы можете изменить комбинацию, переводящую в командный режим, если Вам не по душе клавиши Ctrl+A.

Screen предлагает куда больше возможностей, чем простое создание двух терминалов в одном окне. Например, запустите команду и, не дожидаясь ее завершения, нажмите клавиши Ctrl+A D – *Screen* закроется. Теперь запустите *Screen* снова, на сей раз с опцией повторного прикрепления [reattach]:

```
screen -r
```

Вы снова увидите свой сеанс *Screen*, где будет работать все та же команда. Данная возможность

Наши эксперты

Мы найдем ответы на любой вопрос – от проблем с установкой системы или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа **LXF** – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон
Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Грэм Моррисон
Когда Грэм не обозревает кучи программного обеспечения и не халтурит с MythTV, он готов к ответам насчет любого оборудования и проблем виртуализации.



Валентин Синицын
В редкие свободные минуты главный редактор нашего журнала обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.



Юлия Дронова
Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования www.unixforum.org.

КУДА ПОСЫЛАТЬ ВОПРОСЫ

Пишите нам по адресу: answers@linuxformat.ru **или спрашивайте на форуме:** www.linuxformat.ru

невероятно полезна при работе с SSH. Например, можно зайти из офиса на домашний компьютер, начать закачку чего-то большого и выйти из *Screen*; а загрузка будет продолжаться.

Еще одна удобная для Вас опция – разделение экранов. Она включается клавиатурной комбинацией Ctrl+A, затем Shift+S. Ваш открытый сеанс *screen* со всеми его окнами отъедет в верхнюю половину терминального окна, а нижняя опустеет. Нажатие Ctrl+A Tab переведет Вас в пустую область; нажмите клавиши Ctrl+A C и создайте в ней новое окно. У Вас будет два терминала, видимых одновременно; например, с командой **tail -f** можно просматривать файл журнала и одновременно запускать команды. Man-страница *Screen* объясняет использование этих и многих других опций данной программы. НБ

2 Частный DNS-сервер

В моей домашней сети несколько компьютеров. Как создать частный сервер DNS для поддержки фиктивного домена? Вместо запоминания IP-адресов, я хотел бы обращаться к моим компьютерам по именам наподобие larry.mydomain.home, sally.mydomain.home и т.п. У меня есть выделенный Linux-сервер с Ubuntu 9.10 Server Edition. Сейчас я использую его как сервер для резервного копирования, и на нем работает *BackupPC* (как было описано в LXF125).

Джейсон Ли [Jason Li]

Программа, которая Вам нужна, называется *Dnsmasq* (www.thekelleys.org.uk/dnsmasq/doc.html). Хотя в различных DNS-серверах недостатка не ощущается, данный вариант предназначен именно для той цели, которую Вы себе поставили. Его легко установить, он надежен (я сам много лет пользуюсь им в своей сети) и входит в состав большинства дистрибутивов Linux, включая Ubuntu. *Dnsmasq* работает как сервер, кэширующий имена, и не требует сложной процедуры настройки полномерной службы доменных имен (в отличие от *Bind*).

В простейшем варианте, программа требует информации из двух файлов: **/etc/resolv.conf** и **/etc/hosts**. Первый сообщает, где искать вышестоящие DNS-серверы, и уже предоставляет корректные сведения, если Ваш сервер подключен к Интернету. Файл **/etc/hosts** содержит список имен хостов, их псевдонимы и IP-адреса. Именно так Вы и настраивали бы имена хостов в локальной сети, но благодаря *Dnsmasq* достаточно поддерживать этот файл только на сервере. Формат данного файла – по строке на хост, наподобие

```
адрес имя псевдонимы
```

Например, с учетом ваших имен выйдут такие строки:

```
192.168.1.1 larry.mydomain.home larry
192.168.1.2 sally.mydomain.home sally mail.
mydomain.home
```

Вместо основного доменного имени допускаются псевдонимы, и в данном примере Вы можете обращаться к **larry.mydomain.home** по имени **larry**, а на компьютер **sally.mydomain.home** ссылаться как на **mail.mydomain.home**. Использование доменных имен в соответствии с выполняемыми задачами упрощает организацию и настройку сети. Например, если Вы решите использовать в качестве почтового сервера другой компьютер, потребуется изменить файл **hosts** только в одном месте, а не перенконфигурировать почтового клиента на каждой из машин.

Доменному имени совершенно не обязательно быть фиктивным, так как чтение из файла **/etc/hosts** производится до обращения *Dnsmasq* к серверам имен, перечисленным в **/etc/resolv.conf**. Поэтому Вы можете использовать доменное имя, существующее где-либо еще, и пользователи Вашей сети получат нужный адрес из **/etc/hosts**. Останется только настроить все остальные компьютеры на использование вашего нового DNS-сервера.

Это простейшее из применений *Dnsmasq*. Ваша жизнь еще упростится, если Вы воспользуетесь встроенным в *Dnsmasq* сервером DHCP. Когда одна и та же программа присваивает IP-адреса и обслуживает их для других компьютеров, уже незачем отслеживать все адреса всех компьютеров – можно обойтись их именами. Большинство дистрибутивов по умолчанию настраивают сеть на использование DHCP и отправляют имя хоста в составе этого запроса, так что новый компьютер не нужно настраивать: достаточно добавить строку в конфигурационный файл *Dnsmasq*. Но так как этот файл при обновлении системы может быть перезаписан, лучше отредактировать файл **/etc/dnsmasq.conf**, сняв символ комментария с последней строки:

```
conf-dir=/etc/dnsmasq.d
```

После этого можно добавить в какой-нибудь файл, расположенный в указанном каталоге, примерно следующее:

```
domain=mydomain.home
dhcp-host=larry,192.168.1.1,infinite
```

Первая строка потребуется только один раз. Она задает домен, добавляемый ко всем именам хостов DHCP. Вторая строка выделяет IP-адрес **192.168.1.1** компьютеру **larry** на неограниченный срок. Кроме того, можно указать MAC-адрес сетевой карты компьютера. Определите его, скомандовав **sudo ifconfig** и найдя в выводе команды поле **HWaddr**.

Преимущество данного подхода в том, что адрес и имя хоста для этого компьютера сохраняются неизменными, сколько бы разнообразных дистрибутивов Linux (или даже Windows) Вы на нем ни установили. Формат команды –

```
dhcp-host=01:23:45:67:89:AB,192.168.1.1,larry
```

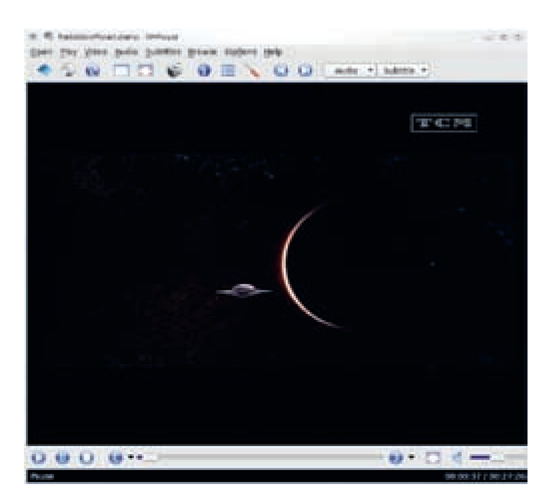
Три поля, разделенные запятыми, задают MAC-адрес, IP-адрес и имя хоста. Если у Вас много компьютеров и Вам требуются напоминания, кто есть кто, добавьте после команды **dhcp-host** строку **txt-record** в следующем формате:

```
txt-record=larry,My desktop computer
```

Первое поле – имя хоста, а второе задает описание, выводимое в ответ на DNS-запрос **txt**. Например, если Вы забыли, для чего у вас компьютер **larry**, дайте такую команду:

```
dig +short larry txt
```

Она освежит Вашу память! МС



› *MPlayer* — популярный проигрыватель, включающий исключительно полезный кодировщик.

3 Слияние AVI-файлов

У меня есть несколько видеофильмов в формате AVI, порезанных на несколько файлов поменьше. Как объединить эти файлы в один AVI? Пытался применить для этого *Cat*, но полученный файл воспроизводится только до конца первой части.

Джон Уотсон [John Watson]

Видеоролики AVI содержат в самом начале файла, перед собственно кадрами, данные о файле: его разрешение, частоту кадров и длительность. Соединив файлы, Вы оставили заголовок первой части без изменений, потому-то и не можете просмотреть весь новый файл целиком. Иногда для слияния файлов рекомендуют применять программу *avimerge*, но тогда случаются проблемы синхронизации звука и видео. Еще один недостаток этой программы – допускается только формат AVI. Существует лучшее решение – *Mencoder*, часть пакета *MPlayer*. *Mencoder* отлично работает и принимает файлы и других форматов, например, MPEG и OGG. Для объединения нескольких файлов дайте следующую команду:

```
mencoder -forceidx -oac copy -ovc copy -o
movie.avi movie1.avi movie2.avi ...
```

Опция **-forceidx** велит *Mencoder* перестроить индекс и метаданные файла. Две следующие дают указание передавать аудио и видео без изменений (oac и ovc означают выходной аудио- и видеокодек, соответственно), а **copy** – естественно, копирование. Затем идет имя результирующего файла, а за ним – список объединяемых файлов. НБ

4 Изменение раздела LVM

Спустя много лет пользования другими дистрибутивами, вернулся к Mandriva. И я потрясен! Но из-за небрежности при установке я получил смехотворно маленький раздел, и теперь мне не хватает места. Не проблема, подумал я: ведь я предусмотрел LVM, и теперь просто увеличу пространство для корневого раздела. Однако обнаружилось, что диск разбит на разделы следующим образом:

```
sda1 -MESH rescue partition
sda2 -Windows 7 NTFS
sda3 -boot partition ext4
```

»

```
sda4 -LVM (containing root, home and swap
volumes)
```

За ними следует огромный кусок пустого, свободного места. Мне очень нужно изменить размеры раздела LVM, чтобы заполнить весь диск и получить возможность выделять логические тома. Просто создать новый раздел и добавить его как дополнительный физический том я не могу, так как Mandriva уже исчерпала запас первичных разделов. Я не враг командной строки, но не хотелось бы угробить таблицу разделов, если что-то пойдет не так.

cmcevoy, вопрос с форумов

О Неприятно, что Mandriva по умолчанию выбрала использование первичного раздела, мешая Вам создать новые, но изменить размеры существующего раздела вовсе не сложно. Рискну предположить, что свободное пространство располагается за концом **sda4**. Если начало раздела останется на прежнем месте, можно удалить раздел, а потом создать новый, побо́льше, не затронув его содержимое. Фактически, содержимое останется в неведении и будет «считать», что раздел имеет прежние размеры.

Чтобы изменить раскладку разделов, потребуется загрузиться с дистрибутива Live CD. Для манипулирования разделами я применяю *cfdisk* – эта команда дает более четкое представление информации по сравнению с *fdisk*; но Вы можете взять любое приложение по Вашему вкусу. Удалите раздел **sda4**, но не сохраняйте внесенных изменений. Затем создайте новый раздел, который заполнит все свободное дисковое пространство – ту область, которая ранее не использовалась, плюс область, которая ранее была занята **sda4**; после этого сохраните таблицу разделов.

Теперь перезагрузите Mandriva и убедитесь в том, что LVM по-прежнему работает. Запуск *vgs* и *pvs* покажет, что ничего не изменилось. Нужно только изменить размеры физического тома, чтобы они соответствовали новым размерам раздела, командой

```
pvresize /dev/sda4
```

Все это делается от имени root. После завершения свободное пространство будет доступно Вашей группе томов. Запуск *vgs* в терминальном сеансе покажет, что объем свободного пространства увеличился. **ПХ**

5 Два диска в деле

В У меня есть старый компьютер с двумя дисками SATA объемом по 500 ГБ и привод DVD-RW. Бывший владелец установил на нем Ubuntu Server 7.04. Мне хотелось бы создать новую инсталляцию 9.10. Как разбить два диска, имеющихся на этом компьютере? Во-вторых, хотелось бы получить совет по настройке *rsync*, чтобы резервное копирование данных с первого диска выполнялось на второй. Данные на первом диске будут резервными копиями содержимого двух других компьютеров, подключенных к моей сети (на одном из них 64-разрядная Windows Vista, а на другом — Linux Mint 8), желательно выполненными *BackupPC*, как описано в LXF125.

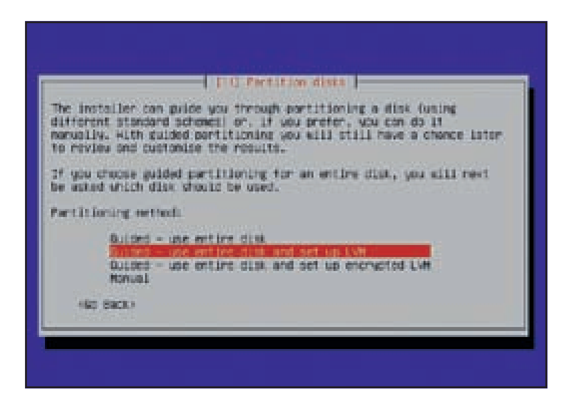
Рональд Зек [Ronald Zec]

О Спросив 10 экспертов по Linux о том, какая схема разбиения на разделы лучше, Вы, скорее всего, получите 10 разных ответов. Если Вы хотите использовать второй жесткий диск для резервного копирования содержимого первого, то все должно быть установлено на первом диске. На сервере большая часть данных хранится в каталоге **/var**, но для *BackupPC* нужен отдельный раздел. Если Вы не запускаете рабочий стол и все ассоциированное с ним ПО, то Ваш корневой раздел не обязательно должен быть большим.

Основное решение, которое Вам нужно принять на данном этапе – сколько пространства выделить файлам и сколько – *BackupPC*. В качестве отправной точки неплоха такая конфигурация:

```
/ 8GB
swap 2GB
/var 200GB
/var/lib/BackupPC 290GB
```

Обратите внимание, что когда Вы назначите схему разбиения диска на физические разделы, изменение этой конфигурации в будущем займет много времени. По этой причине рекомендуется использовать Logical Volume Management (LVM) и разбить диск так, чтобы его единственный физический раздел был поделен на виртуальные тома. Каждый том можно сделать достаточным для Ваших текущих потребностей, а остальное пространство физического раздела оставить невыделенным. Затем, решив, что одна из Ваших файловых систем требует большего объема дискового пространства, Вы сможете просто увеличить ее размер. Инсталлятор Ubuntu Server дает возможность включить LVM в процессе установки системы.



› Linux Volume Management или LVM предоставляет гибкие возможности по управлению разделами.

6 Экранное разрешение

В Я использую три компьютера, подключенных к переключателю KVM (не гипервизору!), и они хорошо работают с Windows XP, а также с Ubuntu 9.04 и более ранними версиями. А вот с Ubuntu 9.10 и Fedora 12 возникла проблема (Fedora 12 я попробовал в надежде ее решить). Создается впечатление, что ни один из этих новых дистрибутивов не умеет читать блок EDID, и экранное разрешение не установить выше чем 800 × 600. Файла xorg.conf нет, и я сгенерировал его следующей командой:

```
Xorg -configure :1
```

Эту рекомендацию я нашел в LXF117. Кроме того, я попробовал пользоваться *Modeline*, как было описано в указанном номере журнала, но X-сервер перестал запускаться, и *Modeline* пришлось удалить. Попытался добавить следующие строки в раздел «Screen»:

```
SubSection "Display"
Modes "1280x800"
EndSubSection
```

— но они, похоже, были проигнорированы. Как добиться, чтобы X-сервер работал с высокими экранными разрешениями, если он не читает EDID (через переключатель KVM) и игнорирует настройки в xorg.conf?

Роб Соломон [Rob Solomon]

О X-сервер игнорирует заданные Вами настройки, так как думает, что Ваш монитор не поддерживает разрешения выше 800×600. Быстро решить проблему поможет грубый трюк: подключите монитор напрямую к компьютеру и проверьте, может ли он читать информацию EDID и корректно настраивать дисплей. Затем дайте команду

```
sudo X -configure
```

Она запишет существующую работоспособную конфигурацию в файл **/root/xorg.conf**. Проверьте, что эта конфигурация работает – выйдите из *X*, войдите в виртуальную консоль и дайте команду

```
sudo X -config /root/xong.conf
```

Если это сработает, скопируйте файл **/root/xorg.conf** в **/etc/X11**, и Вы получите работоспособную конфигурацию, где вся необходимая информация находится в **xorg.conf**, а читать данные EDID с самого монитора будет незачем.

Повторите эту процедуру для каждого компьютера, затем переподключите их через KVM.

Если Вы хотите использовать второй диск для резервного копирования, неплохо будет настроить его идентично первому. Тогда при отказе первого диска Вы сможете подменить его вторым. Установите для второго диска те же файловые системы, но смонтируйте их под другим именем, например, **/backup**. После этого Вы сможете создать резервные копии файловых систем **/var** и *BackupPC* с помощью следующих команд:

```
rsync -ax --delete /var/ /backup/var/
rsync -ax --delete /var/lib/BackupPC/ /backup/
var/lib/BackupPC/
```

Однако эта система не идеальна. Например, она не поможет, если Вы удалите не тот файл и не догадаетесь об этом до того, как выполните очередное резервное копирование. Альтернативный вариант – применять утилиты вроде **rdiff-backup**, они позволяют восстанавливать более старые версии файлов. Но если Вы пользуетесь *BackupPC*, лучше применять ее для резервного копирования всего содержимого диска, а затем использовать *rsync* для переноса на другой диск.

При выполнении резервного копирования на другой диск на том же самом компьютере следует иметь в виду еще несколько фак-

тов. Факт первый: аппаратный сбой, особенно блока питания, выведет из строя оба жестких диска. Гораздо безопаснее использовать диск, подключенный к любой другой точке, хотя бы к порту USB на том же компьютере. Факт второй: всегда найдутся люди, которые завопят о применении RAID. Но RAID – не альтернатива резервному копированию, а просто другой способ защиты. Так или иначе, если Вы все же собираетесь использовать два внутренних жестких диска и синхронизировать их, RAID действительно обеспечивает уровень защиты, аналогичный тому, что Вы получаете при частой синхронизации, если для резервного копирования применяется *BackupPC*.

На основании того, что Вы рассказали, я бы разместил операционную систему и данные – предпочтительно с использованием LVM – на первом диске, и применял бы *BackupPC* для резервного копирования на второй. **НБ**

7 Софт-модемы и ВМы

В Я купил новый компьютер и установил на него Mandriva 2010, Linux Mint 7 и Windows XP. Однако его программный модем [softmodem] Agere Systems PCI не работает в Mandriva 2010. Я как-то возился с настройкой такого модема при установке Linux, но это было несколько лет и пару компьютеров тому назад, и я ничего не помню, кроме того, что для определения чипсета запускалась специальная программа, а после загрузки драйвера пришлось «покувыркаться» с командной строкой. Не могли бы Вы подсказать мне направление, куда двигаться? Я работаю в Папуа, Новая Гвинея, у меня очень медленное модемное соединение, очень высокие расценки на скачивание, и я не хотел бы долго раскапывать эту информацию в Web.

Еще одна проблема — я пользуюсь *VirtualBox*, но не могу организовать общий доступ к данным на моей виртуальной машине Windows XP в установленных дистрибутивах Linux. Нельзя ли как-нибудь скопировать файлы на общий раздел? USB на *VirtualBox* тоже не работает, так что файлы не переместить и таким путем. Как активизировать USB?

Дэрил Брентон [Daryl Brenton]

О Местом Вашей «первой остановки» должен быть сайт http://linmodems.org: там описано все, что необходимо знать об использовании программных модемов, известных также под названием «модемов Windows» [winmodems]. Сайт ориентирован на пользователей, выходящих в Интернет по коммутируемым каналам, он мало использует графику, и должен загружаться очень быстро. На нем имеется прямая ссылка для скачивания утилиты *ScanModem* – скорее всего, это и есть нужный Вам инструмент: http://132.68.73.235/linmodems/index.html#scanModem. Коммутируемый доступ и медленный, и стоит дорого; попробуйте вариант с широкополосным мобильным доступом 3G. Даже при мобильном сигнале среднего качества Вы получите производительность выше, чем у телефонного соединения, и это будет более рациональным вложением средств. Большинство провайдеров используют один и тот же тип модема (Huawei), разве что иногда с другой надписью, и Linux поддерживает эти устройства вполне прилично.

По второму вопросу: виртуальные машины отделены от операционной системы машины-хоста, и обычно не имеют прямого доступа к дискам компьютера или других виртуальных машин. Однако в сети они видятся как самостоятельные компьютеры; поэтому лучше всего предоставить в общий доступ Ваш системный раздел Windows XP (обычно – диск C:). Щелкните правой кнопкой мыши по значку Мой компьютер [My Computer] и выберите из меню команду Общий доступ [Sharing], а затем примонтируйте этот диск в хост-среде Linux. Большинство дистрибутивов Linux и вариантов рабочего стола имеют обозреватель, позволяющий обнаруживать тома Windows, предоставленные в общий доступ, и монтировать их с других компьютеров, расположенных в Вашей сети.

Общий доступ, организованный таким образом, должен работать без добавочного ПО. Но чтобы сделать то же самое из Linux-хоста для гостевой операционной системы Windows, потребуется установить и сконфигурировать на Linux-хосте *Samba*. Соответствующий учебник можно найти в **LXF117** (http://wiki.linuxformat.ru/index.php/LXF117:network).

Наконец, *VirtualBox* позволить Вам активизировать и использовать устройства USB, но только если у Вас есть двоичный вариант сборки – поддержка USB была исключена из версии с открытым кодом. Проверьте, какой вариант дистрибутива установлен у Вас, потому что, возможно, придется удалить его и заново установить из другого источника. Это может быть другое имя пакета, другой репозиторий или версия, доступная для загрузки с сайта www.virtualbox.org/wiki/Downloads. **МС**

8 Дистрибутивы не равны

В В настоящий момент я осваиваю Linux, и мне очень нравится экспериментировать с разными дистрибутивами. Недавно я без всяких проблем создал на своем ноутбуке конфигурацию с двойной загрузкой — Windows 7 и Linux Mint, и все работало замечательно. Прочитав обзоры в вашем журнале, я решил попробовать OpenSUSE и установил этот дистрибутив поверх Mint. Но, похоже, на моем ноутбуке (Dell Latitude D531, доволь- »

Часто задаваемые вопросы

rsync и Unison

» Что это за программа *rsync*, о которой все болтают?

Это способ синхронизации содержимого двух директорий, причем обе будут полностью идентичны.

» А что, нельзя применить для этого *cp*?

cp копирует все подряд. *rsync* – только те файлы, которые различаются. Если изменились большие файлы, копируются только измененные части.

» Как ею пользоваться?

```
rsync --archive --delete /путь/к/источнику/ /путь/к/назначению/
```

создаст точную копию первой директории во второй. Параметр --delete удаляет файлы, которых нет в первой директории, а --archive копирует также права доступа и временные отметки. Замыкающие слэши важны для *rsync*: они указывают, что вы хотите синхронизировать содержимое директорий. Без слэшей программа просто скопирует одну директорию в другую.

» Можно ли синхронизировать директории разных компьютеров?

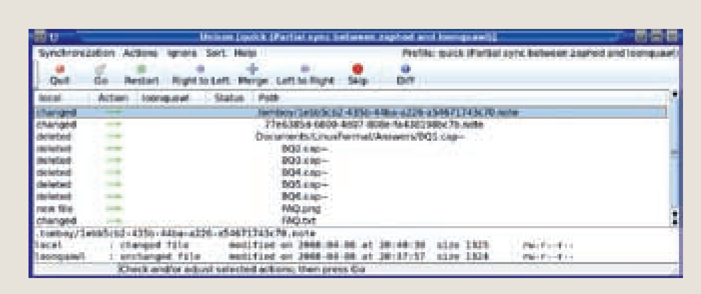
При наличии SSH-доступа к web-серверу, можно сделать так:

```
rsync --archive --delete имя_сервера:/путь/к/сайту/
```

Если один из путей или оба содержат имя компьютера-хоста, *rsync* использует для передачи удаленную оболочку. По умолчанию это SSH, но можно заменить ее любой другой.

» Что если я изменю файлы на двух компьютерах и захочу обновить оба?

Вам необходима программа, разработанная специально для этой цели, например, *Unison* (http://bit.ly/1Symfr). *Unison* использует *rsync* и SSH, так что вы сохраните все вышеперечисленные преимущества; но при этом предназначается для двусторонней синхронизации. Если вы изменили несколько файлов на одном компьютере и еще несколько на другом, то *Unison* гарантирует, что на каждом компьютере у вас будет храниться новейшая версия.



› ***Unison* синхронизирует содержимое двух компьютеров.**

» А если я отредактирую один и тот же файл на обоих компьютерах?

Unison не может читать ваши мысли и определять, какую версию вы хотите сохранить. Но программа сообщит вам, что изменились оба файла, и спросит, какую из версий вы хотите использовать.

» Небось надо заучить сотни опций командной строки?

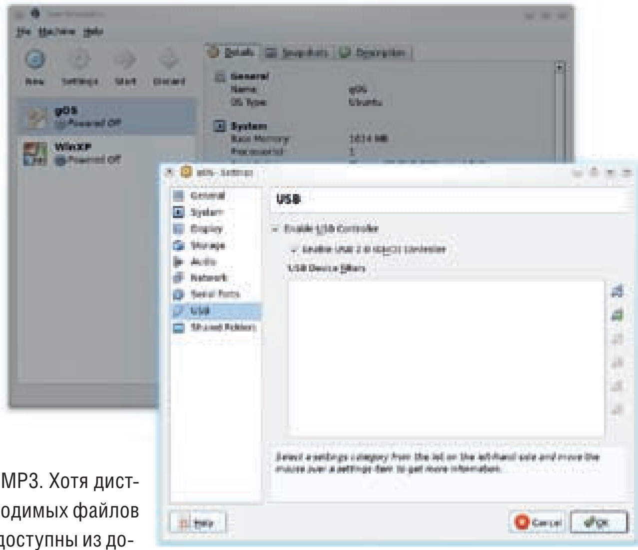
Да, надо. Или пользуйтесь графическим интерфейсом. Он нужен только на компьютере, инициирующем передачу, поэтому вы по-прежнему сможете применять программу для синхронизации с удаленной машиной, где есть *Unison*.

но распространенная модель) нет драйверов беспроводной сети. Попытка поисков в Интернете тоже оказалась неудачной. Выходит, старая версия Mint работает хорошо, а вот новая хваленая версия OpenSUSE — нет. Правильно я понимаю? А если да, то в чем дело — я думал, что основная идея была в том, чтобы абсолютно все было открытым и свободно доступным. Так почему парни из команды OpenSUSE не приложили рук к вещам, которые существуют давно, известны всем и считаются полезными? Конечно, можно вернуться обратно к Mint (не исключено, что так я и поступлю), но не дадите ли вы мне какой-нибудь совет, пока я не принял окончательное решение?

Роб Дэйвенпорт [Rob Davenport]

Одна из причин, по которым не все дистрибутивы поддерживают одно и то же оборудование, кроется в доступности драйверов и условиях лицензирования. Поставщики крупных коммерческих дистрибутивов, к которым относится и SUSE, должны соблюдать осторожность, включая что бы то ни было в состав системы. Если драйвер не абсолютно свободен (т.е. не является открытым и имеет ограничения на распространение), они могут не поместить его в репозиторий. Именно поэтому Linux Mint имеет две версии: одну – для загрузки, и вторую, несколько более ограниченную – для коммерческого распространения.

Загружаемая версия содержит ПО, которое может быть не полностью свободным – а то и нелегальным в стране вашего проживания. То же относится и к таким вещам, как воспроизведение зашифрованных DVD или даже файлов MP3. Хотя дистрибутивы и не включают необходимых файлов непосредственно, они обычно доступны из дополнительного репозитория через менеджер пакетов конкретного дистрибутива.



VirtualBox хорошо работает с устройствами USB, но только если вы используете двоичную сборку.

В Вашем случае, скорее всего, проблема в беспроводной сетевой карте Broadcom BCM94311MCG. Для проверки запустите в терминале от имени root командой **lspci**. Хотя ядро включает драйвер b43 для этого устройства, ему также необходим файл прошивки, а тот в состав дистрибутива по названным причинам не входит. Чтобы решить этот вопрос, подключите свой компьютер к сети с помощью кабеля Ethernet, откройте терминальный сеанс и дайте команду

```
sudo /usr/sbin/install_bcm43xx_firmware
```

Файл прошивки загрузится и установится. Возможно, для появления беспроводного подключения придется перезагрузиться. **ГМ**

9 Окно в Интернет

На моем компьютере установлена 64-битная ОС Ubuntu 9.04 и виртуальная машина *VirtualBox*, в которой работает ОС Windows XP SP2. На компьютере с Ubuntu есть подключение к Интернету. Как средствами Ubuntu соединить два сетевых интерфейса (виртуальный vboxnet0 и реальный eth0) в сетевой мост, чтобы Интернет был и в Windows XP? Есть ли другой вариант попроще?

Rewer8

Разумеется, есть: Вам просто нужно выбрать в качестве типа виртуального сетевого интерфейса *VirtualBox* (Сеть > Тип подключения) «NAT». В результате Windows в гостевой машине окажется примерно в такой же ситуации, как настоящий домашний компьютер за Wi-Fi-маршрутизатором.

Организовать мост также нетрудно: для этого нужна команда **brctl**, обычно входящая в состав пакета *bridge-utils*. Создайте мост командой **sudo brctl addbr br0**, затем добавьте в него интерфейсы (**sudo brctl addif br0 eth0** и **sudo brctl addif br0 vboxnet0**). Переведите сетевые интерфейсы в неразборчивый [promiscuous] режим командами **sudo ifconfig имя_интерфейса 0.0.0.0 promisc up** и настройте **br0** как обычную сетевую карту. Однако в большинстве случаев NAT будет оптимальным решением. Единственная ситуация, с которой он не справится – это предоставление прямого интернет-доступа к сервисам в виртуальной копии Windows XP, что само по себе не лучшая идея. **ВС**

Коротко про...

Top

Случалось ли вам обнаружить, что ваш компьютер тормозит, утратил отзывчивость или «стрекочет» жестким диском, и недоумевать о причинах такого поведения? В следующий раз запустите в терминале *Top*. Эта программа предоставляет большой объем информации в очень сжатом формате.

Верхняя часть экрана отображает системную статистику: время работы компьютера, число пользователей, количество выполняющихся задач и т.д. Далее, программа выводит общую загрузку процессора, распределенную между пользовательскими программами (**us**), системными операциями (**sy**), программами, сменившими свой приоритет (**ni**), простаивающими (**id**) и процессами, находящимися в ожидании (**wa**). Очевидно, простаивающие процессы относятся не столько к использованию, сколько к неиспользованию, и это число показывает доступный объем свободных ресурсов процессора. Число, относящееся к ожидающим программам, показывает процессы, которые задерживаются из-за ожидания наступления какого-нибудь другого события – обычно ввода/вывода на диск.

Ниже приведен список работающих задач, отсортированный по нагрузке на процессор. Он покажет, которая программа забирает все циклы CPU. Нажмите клавиши Shift+F, чтобы изменить сортировку этого списка по использованию, например, памяти.

Top покажет, какая программа отнимает сколько ресурсов памяти и процессора.

Top не только показывает процессы, но и способен менять их. Например, чтобы завершить процесс, нажмите клавишу K [kill]; а чтобы сменить приоритет процесса, нажмите клавишу R [renice]. Увеличение значения nice [«вежливости»] для программы делает ее уступчивее по отношению к другим процессам («только после вас») и менее алчной до циклов CPU. Команды **kill** и **renice** требуют указать идентификатор процесса (process ID, PID), как показано в левом столбце. Если вы укажете неверный идентификатор, это чревато катастрофой. Но при запуске *Top* в X-терминале можно дважды щелкнуть мышью на PID нужного Вам процесса и нажать среднюю кнопку мыши, чтобы скопировать его и избежать недоразумений.

Top предлагает намного больше опций, к большинству из которых можно получить доступ нажатием единственной клавиши. Самая важная клавиша – H [help], она отображает онлайн-справку о программе.

Большой вопрос Как использовать телефон под управлением Android в качестве модема 3G?

В У меня есть мобильный телефон, работающий под управлением Android, и «безлимитный» пакет 3G. Как я могу этим воспользоваться, чтобы подключить свой ноутбук к Интернету и не переплачивать за другой пакет и 3G-модем?

Мартин Уэллс [Martin Wells]

О Большинство смартфонов умеют работать модемами; обычно для этой цели требуется соединение через USB или Bluetooth. В стандартной системе Android это невозможно, потому что для этого необходим доступ с привилегиями root. Однако доступ с привилегиями root к телефону Android можно получить без особых проблем, как показывает быстрый поиск в Web. Единственное, о чем умалчивают web-страницы – это проблемы с безопасностью из-за полного доступа root ко всем приложениям, работающим на телефоне. Серьезно подумайте, то ли это, что Вам нужно, потому что существует и альтернатива: программа *Proxoid*, которая превращает Ваш телефон Android в прокси-сервер Web. Набор разработчика Android (Developer Kit) содержит ПО для отправки TCP-запросов на телефон через USB-кабель.

На своем телефоне установите *Proxoid* через Android Market. Чтобы телефон поддерживал связь с компьютером через USB, надо активизировать отладку USB. На домашнем экране выберите команды Меню > Настройки > Приложения > Разработка > [Menu > Settings > Applications > Development] и включите опцию Отладка USB [USB debugging]. Затем запустите и активизируйте *Proxoid*. На компьютере, чтобы он распознал Ваш телефон, нужно выполнить настройку: подключите устройство через USB, скомандуйте

```
sudo lsusb
```

и поищите запись для своего телефона (если Вы не уверены, запустите команду дважды: при подключенном и отключенном телефоне). Вы увидите идентификатор (ID); за ним идут два шестнадцатеричных числа – идентификаторы производителя и продукта. Так, у нас это были

```
Bus 001 Device 003: ID 22b8:41db Motorola PCS
```

Затем создайте файл **/etc/udev/rules.d/11-android.rules** от имени root и добавьте в его состав следующую строку:

```
SUBSYSTEMS=="usb", SYSFS{idVendor}=="22b8",
SYSFS{idProduct}=="041db", MODE="0666",
OWNER="ваше_имя_пользователя"
```

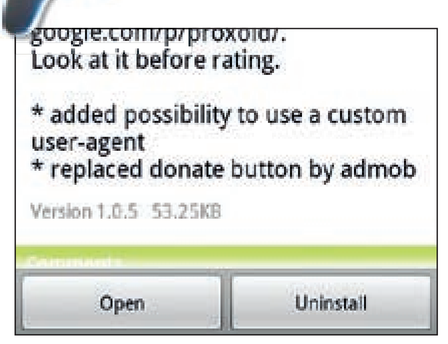
Замените эти коды на соответствующие Вашему телефону и укажите свое имя пользователя. Теперь при подключении телефона владельцем узла устройства, создаваемого *udev*, будете Вы.

Затем потребуется установить Android SDK. Если он есть в репозитории Вашего дистрибутива, добавьте его стандартным образом; если нет, загрузите пакет с адреса http://developer.android.com/sdk – это файл **.tar.gz**. Распакуйте ПО где-нибудь в пределах Вашего домашнего каталога, запустите терминал, перейдите в каталог **android-sdk-linux_86/tools** и дайте команду

```
./adb forward tcp:8080 tcp:8080
```

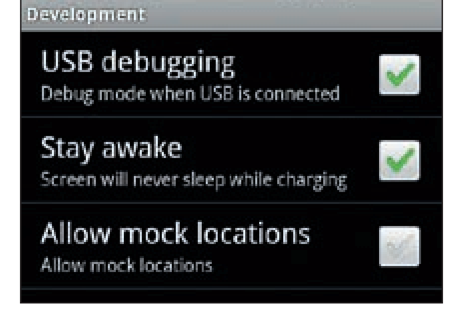
Запрос на TCP-соединение будет перенаправлен через порт **8080** компьютера на порт **8080** телефона, где по умолчанию работает *Proxoid* (при желании, эту настройку можно изменить средствами *Proxoid*). Теперь остается только настроить Ваш браузер для использования прокси-сервера на порту 8080 локального хоста, после чего все его запросы будут проходить через телефон. Когда Вам потребуется переключиться обратно на обычное интернет-соединения, просто отключите прокси в Вашем браузере.

Шаг за шагом: Настройка модема Android



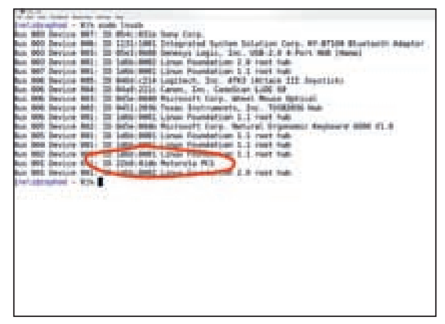
1 Установка Proxoid

Первый шаг – установка приложения *Proxoid*. На своем телефоне найдите *Proxoid* в репозитории Android Market и установите его.



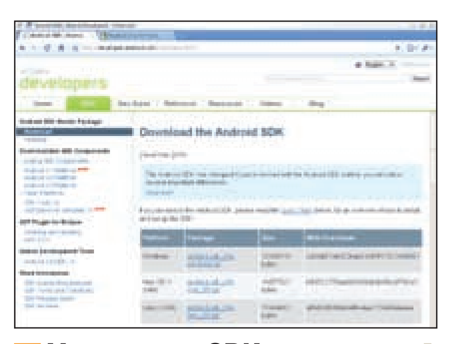
2 Активизация отладки USB

Далее, выберите на телефоне опции Applications > Development и активизируйте отладку USB.



3 Настройка udev

Перейдите на свой компьютер и воспользуйтесь командой **lsusb**, чтобы найти идентификаторы вашего телефона. Затем создайте правило *udev*, чтобы распознать его при следующем подключении.



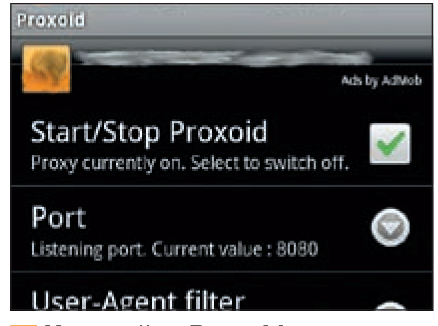
4 Установите SDK

Воспользуйтесь менеджером пакетов либо возьмите tar-архив и распакуйте в удобном месте (точное местоположение роли не играет).



5 Настройка браузера

Настройте браузер так, чтобы в качестве прокси-сервера использовался **localhost:8080**, но вашу локальную сеть отсюда следует исключить.



6 Настройка Proxoid

Параметры по умолчанию вполне годятся, однако *Proxoid* можно и перенастроить – например, указав для прослушивания другой порт. LXF

Лучшие в мире новинки свободного ПО

LXFHotPicks



Ник Вейч
Ник компилирует Hotpicks, используя только текстовый редактор и головной мозг. А иногда — только текстовый редактор.

Генеалогическое ПО

Gramps

Версия 3.1.3 **Сайт** www.gramps-project.org

Вот люди! И с ними тяжко, и без них никак, согласно поговорке. Но одно можно сказать наверняка: если бы не они, то и нас бы не было.

За последнее десятилетие генеалогия стала популярной как никогда, в основном потому, что на серверах по всему миру рассеяно немало информации почти о каждом. Не нужно больше лезть в пыльные архивы и беспокоить почтенных очкастых старцев, шагнувших за восьмидесятилетний рубеж. Но данные, неважно откуда почерпнутые, надо организовать в некую презентабельную форму. Какой смысл прилагать столько усилий, если потом нельзя изобразить симпатичное древо?

Тут-то и приходит на выручку *Gramps*. Импозантные схемы – это его конек, но ими дело не ограничивается. *Gramps* скорее являет собой генеалогический органайзер: с одной стороны, база данных, с другой – программа для записей, он не только помогает вам проследить связи между вашими злодейскими предками, но и тщательно подшивает все ссылки, фото и прочие данные о пращурах. Короче говоря, он не ждет, когда вы зальете в него информацию, он помогает вам изучить ее. Однако это означает, что и ваш сбор информации должен быть более организованным.

> «Импозантные схемы — его конек, но ими дело не ограничивается.»



› Это обычный развернутый DXF-примитив, но вы вообразите гигантскую головищу с отверстиями для глаз.

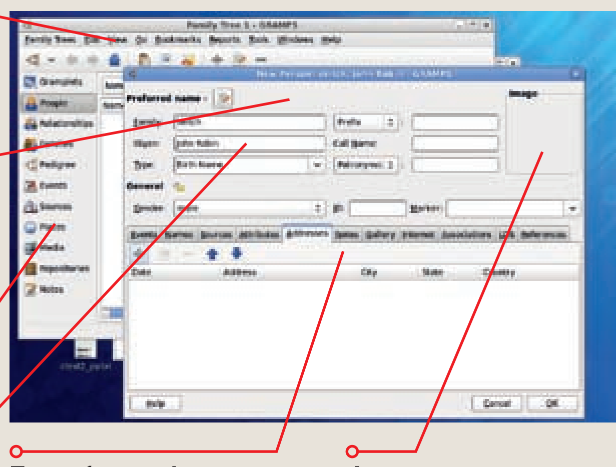
Будьте терпеливы

Gramps рассчитывает, что вы будете регистрировать «репозитории» своих источников информации, так что, вознамерившись воссоздать развесистую родословную, не ждите, что на это хватит пяти минут; но зато вам гарантируется, что со всеми перекрестными ссылками, ваше генеалогическое древо будет действительно солидным и достоверным.

Для обеспечения совместимости с другими программами и с сайтами, *Gramps* поддерживает формат GEDCOM и разные типы XML, которые можно импортировать и экспортировать, что позволяет делиться информацией с другими (а еще *Gramps* работает на Windows и на Mac OS, благодаря Python).

И если вдруг – поскольку предполагается, что большинство жителей Европы в состоянии это сделать – вы сумеете проследить своих предков вплоть до Альфреда Великого [король Англии, живший в конце IX в., – *прим. пер.*], то уж вышлите нам распечатку своего фамильного древа.

Исследуем Gramps



Фамильное древо
Это – главное окно фамильного древа для текущей базы данных.

Записи
Каждая добавляемая личность получает свою запись, и в нее можно заносить подробности.

Источники
Панель инструментов дает быстрый доступ ко всем носителям и источникам информации.

Личные сведения
В отличие от некоторых подобных программ, здесь, если вы не уверены в показаниях, можно делать догадки.

Подробная информация
В запись каждого человека, кроме основных данных, можно добавлять любую информацию.

Фото
Gramps поддерживает множество типов мультимедиа, и вы можете добавлять фото своей родни.

Текстовый редактор

Tea

Версия 26.2.2 **Сайт** http://tea-editor.sourceforge.net

Нехватки текстовых редакторов в этом мире не наблюдается. Тем не менее, нам присуще являть великодушие и поддержку ко всем новым представителям этого класса: вдруг окажется истинным бриллиантом? Похоже, *Tea* весьма близко подплыл к береговой линии программирования, накопив солидную поддержку выделения синтаксиса для изрядного стога языков. На данный момент среди них HTML и C/C++, а также Fortran, PHP, Java, Lua и охапка прочих, о которых вы, честно говоря, наверное, и сроду не слыхали.

В несложном интерфейсе используются вкладки – и при работе с документами (чтобы разделять разные файлы), и сбоку, для переключения между разными функциями программы. Здесь имеются встроенный *MC*-подобный файловый менеджер, всяческие настройки, а также руководство пользователя.

Хотя *Tea* быстр и весьма прост в использовании, он вряд ли отвадит нас от программ вроде *Kate*, *Vim* или *Emacs*; и все же тут имеется несколько экзотических инструментов, не лишенных полезности.

Точка, точка, тире

Несомненно, *Tea* – единственный из всех известных нам текстовых редакторов, в котором имеется пункт меню для перевода азбуки Морзе на английский и русский и обратно. Среди прочих встроенных функций – быстрая работа с разными типами списков, перевод чисел из одной системы в другую и анализ частей текста или всего текущего документа полностью.

Для заварки чая нужен чайник. Для компиляции *Tea* нужен обычный триумвират – **qmake**, **make** и затем **make install** [в оригинале – игра слов: tea – чай (англ.), – *прим. пер.*] Поскольку это – приложение *Qt 4*, надо будет установить соответствующие библиотеки, в том числе и версии для разработчиков. В наши дни в управление пакетами вернулся здравый смысл, так что большая часть необходимого должна найтись в пакете **Qt4-devel**, или подобном, в зависимости от вашего дистрибутива. Больше ничего особенного не требуется, а поскольку приложение очень простое, то и места на диске оно практически не займет.

Конечно, многого пока не хватает, но эта симпатичная программа таит немало приятных сюрпризов. И нам особенно приятно заметить, что ее автор – Петр Семилетов – был гостем на страницах LXF.

›Напишите пару оскорбительных замечаний про Майка — *Tea* переведет их в азбуку Морзе, и цензоры ничего не поймут.

> «В нем имеется перевод азбуки Морзе на русский и обратно.»

Программа для чтения новостей

Liferea

Версия 1.6.2 **Сайт** http://liferea.sourceforge.net

На заре существования Всемирной паутины кто-то запустил идею «подписки» на сайты. Она заключалась в том, что если вам нравится сайт, то вы, вероятно, захотите узнавать об изменениях, происходящих на нем. Вполне справедливо. А потом настал сдвиг парадигмы – есть у парадигм такой прискорбный обычай.

Мир перешел на синдикацию новостей: в определенном месте сайта ведется журнал изменений, чтобы клиентские программы считывали его и сообщали вам об изменившемся содержимом. Естественно, ломались копья по поводу формы подачи этих новостей, и в результате теперь есть три способа: Atom, RSS и RSS2. Однако это не важно, поскольку любая программа чтения новостей, для которой не жаль места на диске, может обрабатывать их все, и *Liferea* – не исключение.

Программ для чтения новостей немало, но эта, основанная на *GTK*, определенно стоит внимания. Рассматривая минимализм как высшее проявление эстетики, она все же умудряется включить в простой интерфейс немало функций.

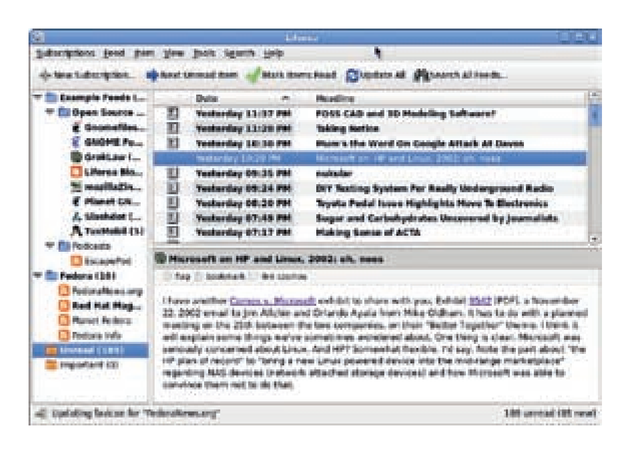
Читая Google

Liferea поставляется с готовым набором встроенных новостных лент, что должно обрадовать пользователей Linux, но, что важнее, в ней очень легко добавлять и управлять новыми источниками информации. Одна из приятных новых функций – синхронизация с учетной записью Google Reader: просто введите свое имя пользователя и пароль, и все ваши новости волшебным образом появятся в *Liferea*.

Добавив подписки, вы можете управлять ими, просто перетаскивая их на левую панель, и, при желании, создавая папки, чтобы собирать в них информацию.

Среди плюсов *Liferea* – малое количество зависимостей (если вы устанавливаете ее в среде Gnome), и она идеальна в качестве легкой программы для чтения новостей на рабочем столе.

Версия, которую мы тестировали – мелкое исправление ошибок предыдущего релиза, доступного в репозиториях дистрибутивов. Если вам интересно познакомиться с *Liferea*, мы бы советовали взять предыдущую версию и после выхода пакетов с исправлением обновить ее.



›Простой интерфейс Liferea упрощает добавление новостей и управление ими, и теперь в ней есть поддержка Google Reader.

> «Одна из новых функций — синхронизация с Google Reader.»

Web-платформа

Hop

Версия 2.0 **Сайт** http://hop.inria.fr

Знакомы ли вы с диффузным Web? Если нет, вас никто не посмеет обвинить, так как этим термином оперируют несколько вольно, когда речь заходит об облачных технологиях мира онлайн.

В глазах создателя *Hop*, диффузный Web – это мир, где пользователи взаимодействуют с web-сервисами, различными способами применяя стандартные протоколы: например, через привычный браузер, мобильный телефон или планшет. Он делает заключение, что многие из этих сервисов могут быть локальными – возможно, вашим домашним web-сервером или медиа-плейером.

Если вы верите в будущее Web, то с *Hop* стоит познакомиться. Назвать его языком программирования явно недостаточно – это скорее среда разработки для создания быстрых и совместимых со стандартами web-приложений для небольшого или среднего трафика. Сам по себе язык можно использовать для создания апплетов примерно так же, как может применяться, например, SDK для Gnome при создании приложений рабочего стола.

Разница здесь в том, что приложение пишется для работы в браузере. *Hop* использует для обслуживания своих приложений и апплетов собственный сервер, и настроен так, чтобы упростить создание виртуального рабочего стола.

Тесная интеграция

Приложения *Hop* создаются встраиванием кода, который затем генерирует HTML по запросу (и кэширует его, если не предвидится изменений содержимого). Это довольно стандартно для современных динамических web-технологий, но *Hop* извлекает преимущества из тесной интеграции с кодом сервера, позволяющие ему претендовать на бонус в плане производительности. Тестирование этих претензий выходит за рамки HotPicks, но проект производит впечатление стоящего. Что касается сборки сервера, то код легко компилируется с единственной крупной зависимостью – компилятором *Bigloo* (http://bit.ly/hhVOQ), но вам не обязательно устанавливать его, чтобы выяснить, надо ли с ним связываться, поскольку есть демо-сайт, где выложены примеры приложений.

> «Это среда для создания стандартных web-приложений.»



› Демо-приложения дадут представление о том, на что способна система.

Менеджер шрифтов

Font Manager

LINUX FORMAT Hottest Pick

Версия 0.4.3 **Сайт** http://bit.ly/E3Vke

Все, что вы читаете в *Linux Format* или на дисплее вашего любимого компьютера с Linux, доносится до вашего сознания посредством магии перевода невысказанных мыслей в маленькие закорючки. Ваш мозг волшебным образом декодирует эти закорючки, превращая их обратно в слова.

За всю историю существования печати, колоссальное количество дискуссий, усилий и ухищрений велось и было вложено в природу, размер и форму этих загогулин: одни были выбраны благодаря тем чувствам, которые они вызывают – например, доверие или удивление; другие получили путевку в жизнь просто за свою читабельность.

За время, прошедшее с поры Вильяма Кэкстона [William Caxton, английский первопечатник], в компьютеры большинства из нас вселилось несметное число шрифтов, и порой управление ими выливается в проблему. Оно вызывает особенную сложность, если вы имеете привычку создавать печатные документы или разрабатывать графику с буквами – при этом, конечно, полезен широкий выбор шрифтов, но при каждом щелчке 20 минут ждать загрузки WYSIWYG-меню шрифтов вашей программы радости мало. На других платформах эту проблему решает менеджер шрифтов, и теперь она будет так же решаться и для Linux.

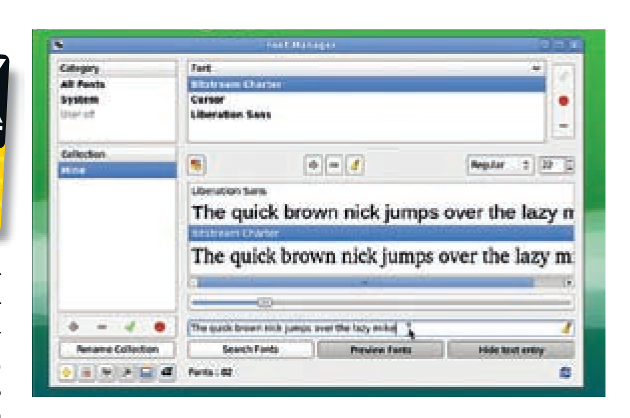
Организуйтесь

Font Manager – одно из таких приложений (аналогичный вариант – *Fonty Python*). Подобные программы находят все доступные в вашей системе типы шрифтов. Вы можете создавать собственные группы шрифтов и по желанию отключать или снова включать их, а также организовывать ваши шрифты в группы согласно проектам, где они используются, или их стилям, или любой иной классификации, которая придет вам в голову.

Это приложение Gnome, но оно должно нормально работать и в других средах рабочего стола, при наличии установленного *PyGTK*. И хотя, возможно, другие приложения и глубже проникли в природу шрифтов, этот простой менеджер эффективен, компактен и удобен.

Вы можете подождать, пока *Font Manager* появится в грядущих обновлениях вашего дистрибутива, или установить его прямо сейчас с LXFDVD.

> «Можете создавать, отключать и снова включать группы шрифтов.»



› Объединяйте свои шрифты в группы с помощью *Font Manager*, чтобы загружать только необходимые для текущей работы.

Мониторинг ресурсов

Iotop

Версия 0.4 **Сайт** http://guichaz.free.fr/iotop

Дело было так: у нас был *Top*, монитор процессов. Потом был *Ntop*, монитор сети, и *Mtop*, монитор базы данных *MySQL*. Intel предоставил нам *Powertop*, превосходный и амбициозный монитор энергопотребления.

А ныне мир свободного ПО подарил нам *Iotop*, который проводит мониторинг, э-э, активности ввода-вывода. На самом деле, этот инструмент много полезнее некоторых других *top*'ов. Ну, может, он и не самый хитовый *top* – этим, наверное, всегда будет *Top*; но явно и не последний среди *top*'ов.

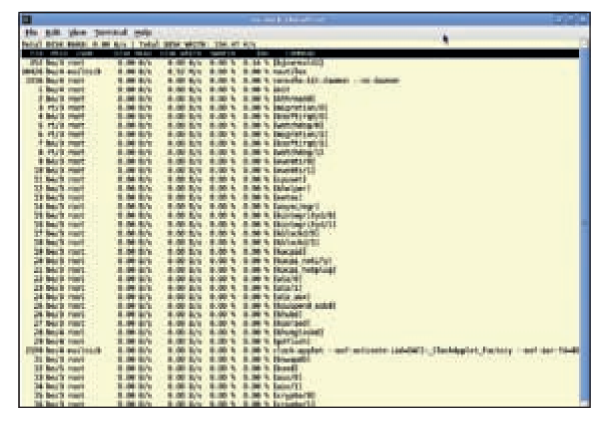
Представьте себе такую картину: вы праздно сидите у компьютера и размышляете, не пора ли отправиться спать – учитывая, что уже пять утра. И вот в тот самый миг, когда вы наводите мышь на кнопку отключения, диск разражается бешеной деятельностью.

В чем дело? Настольный поисковик индексирует ваш диск? Или активизировался какой-то фоновый процесс? А может, ваши данные пожирает вирус? Ах, нет, мы же не в Windows… В любом случае, бывает небесполезно узнать, в чем причина столь бурной активности ввода-вывода: может, лучше бы ее и пресечь.

Проведите проверку

А может, вы просто хотите знать реальную производительность приложения, в котором вы работаете, или оценить интенсивность подкачки. На все эти вопросы ответит *Iotop*.

Взяв пример со всех предыдущих *top*'ов, этот *top* также отображает в консоли список процессов в порядке убывания числа обращений к диску, а также раз-



› Есть и другие способы оценить активность диска, но на фоне других *top*'ов этот дает больше.

> «Может, вы просто хотите знать реальную производительность.»

личную идентификационную информацию, например, ID процесса, имя пользователя и т. п. Здесь есть разумный набор параметров командной строки и информативная man-страница с пояснением их назначения. Данная утилита использует процедуры учета ядра Linux, так что вам потребуется установить ядро версии 2.6.20 или выше, а также Python.

В общем-то даже и неважно, нужен вам мониторинг ввода-вывода или нет – одно сознание, что вы в принципе можете его провести, должно вызывать у вас чувство глубокого удовлетворения. А раз так, стоит установить *iotop* прямо сейчас.

Световой стол

Darktable

Версия 0.4 **Сайт** http://darktable.sourceforge.net

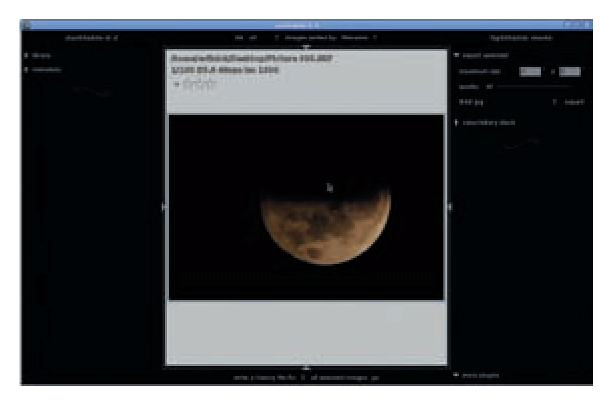
Darktable – любопытный маленький проект, вышедший из недр обработки RAW-изображений. Правда, он припозднился: все же в Linux уже существует низкоуровневая поддержка работы с DSLR RAW-файлами. Некоторые приложения, например, *Digikam* и *F-Spot*, прекрасно отображают и организовывают RAW-снимки. Но инструментов, реально способных обрабатывать данные из RAW-файла и наводить мост между изображениями с высокой информационной плотностью и приложениями, не так уж много (один из них – *Rawstudio*). Да, есть еще загрузчик *UCRaw*, но, честно говоря, на каждый снимок здесь приходится головной боли в избытке.

Похоже, что на *Darktable* повлияли приложения для редактирования и управления изображениями с других платформ. Интерфейс отличается минимализмом и простотой – загрузите его, и перед вами предстанет черный прямоугольник. Меню всплывают при щелчке по любой стрелке на краях рамки, предлагая свою скуповатую щедрость. На данный момент приложение принадлегло на импортирование изображений, предоставив вам возможность поиграть с метаданными и произвести ряд базовых исправлений.

Быстрый оператор

Однако стоит заметить, что *Darktable* построен на архитектуре модулей расширения, так что добавление новых функций должно пойти весьма быстро.

Здесь есть неплохой показ гистограмм, и поскольку изображение обновляется почти что в реальном времени, работать с ним довольно легко. Снимки



› Хотя сейчас как редактор изображений *Darktable* скуден, в будущем его функции могут быть расширены модулями.

> «Похоже, что на него повлияли приложения с других платформ.»

не будут окончательно изменяться или отрисовываться до тех пор, пока вы не решите экспортировать их, что значительно упрощает и ускоряет эксперименты с фотографиями.

И хотя упор сделан на управление, а не обработку изображений, в этой области *Darktable* мог бы порадовать и большим количеством функций. «Инструменты библиотекаря» по работе с огромным количеством изображений несколько ограничены; впрочем, можно возразить, что этим отличаются и коммерческие приложения на других платформах.

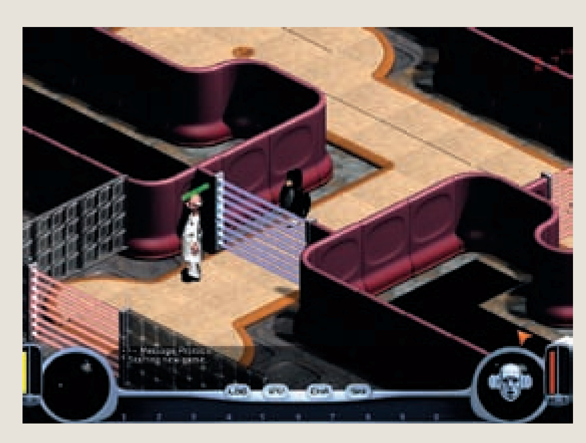
Но если вас занимает цифровое искусство, то это приложение заслуживает местечка в вашем арсенале.

HotGames Развлекательные приложения

Ролевая игра/экшн

FreedroidRPG

Версия 0.13 **Сайт** http://freedroid.sourceforge.net



› «Извините, как пройти на консервный завод?»

Жил-был гений по имени Эндрю Брейбрук [Andrew Braybrook]. Помимо прочих великих деяний, этот титан среди смертных написал игру столь хитроумную, столь дьявольски затягивающую и столь полную роботами, что превзошла она все жанры и все земные концепции величия. И имя ей было *Paradroid*.

Речь сейчас не о ней, но некотором извилистом роде эти две игры связаны. *Freedroid* был переписанной для Linux версией этой классики C64, весьма схожей с оригиналом: игрок захватывает роботов, пытаясь истребить их посредством решения логических мини-игр.

FreedroidRPG – опять-таки не она, а ответвление, преобразованное в изометрическую ролевую 3D-игру в жанре экшн/приключения. Суть в общем та же: люди привлекли на свою сторону пингвина, чтобы сражаться с обезумевшими роботами, намеренными уничтожить человечество. Интерфейс «наведи-и-щелкни» помогает провести персонажа по местности, а по пути решать головоломки, сражаться с роботами, и, кхе, применять магию.

Чокнутый профессор

Как ни странно, но играть в это можно. Когда вы запускаете игру, появляется диалоговое окно с руководством – вы общаетесь с безумным профессором, который вас, собственно, и создал, что вполне ориентирует вас на подвиги. Батальные сцены и некоторые другие места в игре, конечно, нуждаются в доработке, но *FreedroidRPG* выглядит весьма неплохо продуманной игрой.

> «Люди привлекли пингвина помочь им в борьбе с роботами.»

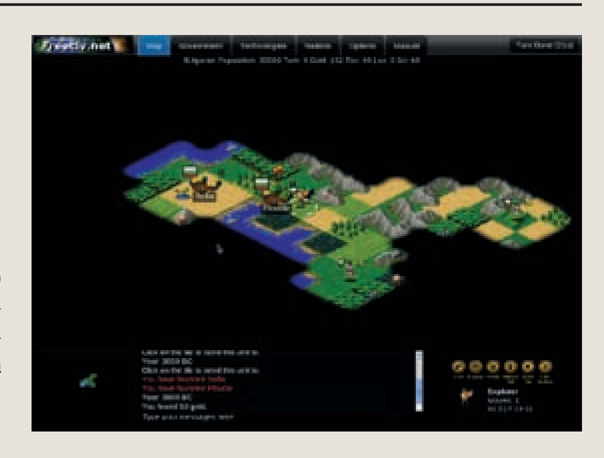
В ней используется графика OpenGL, так что рекомендуем видеокарту с поддерживаемым ускорителем, а не то довольствуйтесь четырьмя кадрами в секунду. Отмечена также проблема со звуком в системах с *Pulseaudio*.

Компиляция из исходных текстов довольно проста, если у вас есть различные пакеты SDL и их соответствующие версии для разработчиков.

Многопользовательская стратегия

Freeciv

Версия 1.1 **Сайт** http://freeciv.net



› Куда бы вы ни приклонили свою шляпу, варвары обязательно попытаются ее у вас стащить.

Civilization от Сида Мейера [Sid Meier] была одной из самых чудесных игр в мире Amiga и прочих компьютеров 16-битной эры. Сочетая элементы настольной игры и компьютерной механики классики Unix, *Empire*, она вполне заслужила все восторги, которые вызвала, и сиквелы, которые породила.

Freeciv – новая интерпретация вечно популярного сюжета – проект, существующий уже несколько лет, но, возможно, пока вы отвлеклись на пару секунд, она успела ускользнуть с рабочего стола, и теперь в нее можно играть онлайн.

Нечестно включать в HotPicks онлайн-игру? Вовсе нет, если она по-прежнему с открытым кодом – и к тому же хорошая. Как и в ее предке, вы можете основать новую нацию, бороться с плохими парнями, исследовать мир – и потом завоевать его. Здесь те же известные боевые единицы и привычные действия, что и в исходном варианте игры. Меню сменились на большие кнопки в верхней части экрана, а набор значков внизу справа дает доступ к управлению войсками.

Многочисленные противники

Постоянно щелкать по ним – не самое веселое занятие, но не волнуйтесь: есть ведь еще и клавиши на клавиатуре. Внизу слева отображается мини-карта, а панель под основной картой используется для сообщений и отображения текущих событий. Возможно, вы сочтете, что подобная сетевая игра несколько громоздка и неуклюжа, но она на диво хорошо работает, благодаря хитроумному JavaScript.

> «Имея исходники, вы всегда сможете подогнать ее по вкусу.»

Помимо игры против других людей, вы можете сражаться и с компьютерными противниками, если вам так больше нравится, и, наверное, это наилучший способ для новичков.

Что касается реализации *Civ*, она вполне играбельна, а имея исходники, вы всегда можете подогнать ее по своему вкусу или просто запустить в локальной сети, чтобы сразиться со своими друзьями, коллегами или домашними животными.

Язык программирования

Freeforth

Версия 1.2 **Сайт** http://christophe.lavarenne.free.fr/ff

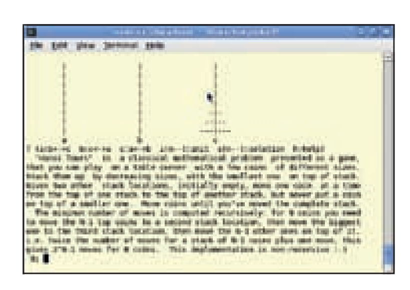
До появления объектно-ориентированных языков программирования на компьютерах работали настоящие языки, как, например, Forth. Если вы застали эпоху Jupiter Ace, вы, возможно, помните этот странный диалект, маскирующийся под высокоуровневый язык. Листинг Forth выглядит так, как будто вы напечатали свой список покупок на перфорационной машинке с кривой головкой. Ну, если вы помните, что это такое.

Forth применяет словарную стековую структуру, поэтому кодовые процедуры могут быть определены, сохранены и исполнены дальнейшими операторами. У него довольно запутанный синтаксис и эксцентричная инверсная польская нотация. Это нечто вроде причудливого сочетания интерпретатора и компилятора – по крайней мере, большинство его версий.

Freeforth не таков, но если вы действительно хотите отыскать инновации, то лучше всего будет пройтись по хорошему руководству. Вот вам Forth в действии:

```
0; 5 6 7 * + . ;
47 0;
```

Вы, небось, изумляетесь, на кой вам вообще знать этот Forth. А это весьма эффективный язык, и не такой низкоуровневый, как машинный код. Его держит на плаву академический интерес, но для него существуют и вполне реальные приложения в реальном мире. На нем программируется масса микроконтроллеров, и полученный в результате код нередко короче и быстрее, чем у более сложных языков типа C. Во *Freeforth* имеются изящные функции, примеры кода и полезные библиотеки.



› Хотя синтаксис и сложен, в *Forth* делаются весьма полезные вещи.

Утилита форматирования кода

Highlight

Версия 2.14 **Сайт** www.andre-simon.de

Мы рассматривали *Highlight* в HotPicks в **LXF99**, но нам пришлось снова обратить на него внимание: он изрядно продвинулся. Теперь есть поддержка для большего количества языков, и добавлены новые функции автоформатирования.

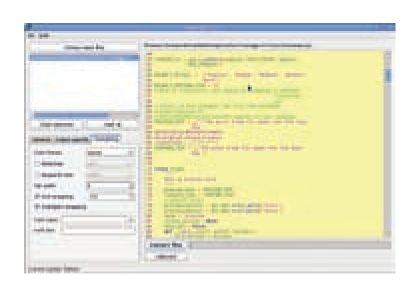
Эти последние облагородят ваш код, трансформировав его в приемлемый формат, например, по схеме, используемой разработчиками ядра Linux.

Highlight может работать как из GUI, так и из командной строки. Первый вариант позволит предварительно просмотреть результат и подправить его. Когда вы определитесь с параметрами – какие использовать, для большого числа файлов командная строка будет быстрее и эффективнее.

Можно менять размер отступа, включать перенос и нумерацию строк и задавать параметры шрифтов. Прочие опции зависят от желаемого продукта на выходе. Помимо исходного HTML, *Highlight* теперь поддерживает форматы XHTML, TeX, LaTeX, RTF, XML и даже SVG, и во многих из них есть специальные настройки.

Кроме форматирования кода, на выходе можно получить подсветку синтаксиса. Для этого имеется множество цветовых схем, включая эмуляцию популярных IDE и текстовых редакторов. Ко всему хорошему, он еще и работает весьма прытко.

Собрать программу из исходных текстов легко: зависимостей немного, вдобавок *Highlight* уже проторил тропку в репозитории большинства дистрибутивов. LXF

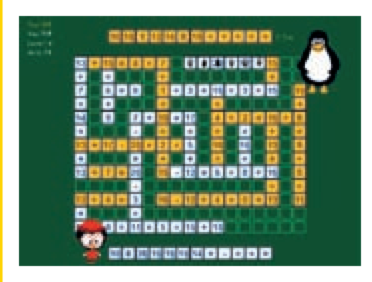


› Даже код Python можно сделать посимпатичнее, вложив в него немного любви от *Highlight*.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

» TuxMath Scrabble 0.7.3
Превращает общение со сложными математическими уравнениями в удовольствие.
http://new.asymptopia.org



› Классическая настольная игра с новыми наворотами.

» BusyBox 1.15.3
Рабочая лошадка небольших дистрибутивов и встраиваемых версий Linux.
http://busybox.net

» Enna 0.4.0
Первый публичный релиз этого нового приложения для медиа-центра.
http://enna.geexbox.org

» Mapnik 0.7.0
Набор инструментов для картографического сервиса со встроенным сглаживанием, чтобы добиться красивых результатов.
http://mapnik.org

» Argyll Color Management System 1.1.0
Откалибруйте свой монитор и печатайте более точные изображения.
www.argyllcms.com

» Pyrad 0.3.1
Необычная, но интригующая программа запуска приложений в стиле «колеса» для KDE 4.x.
http://bit.ly/9jdvpw

» WordPress 2.9.1
Любимый блоггерами движок улучшил обработку и редактирование изображений.
http://wordpress.org

» Fluidsynth 1.1.1
Звуковой синтезатор реального времени, совместимый с SoundFont 2.
http://fluidsynth.resonance.org/trac

» Griffith 0.11
Эта небольшая утилита позволяет создавать базу данных, чтобы работать с коллекцией фильмов.
www.griffith.cc

На диске

Настоящий Unix и куча программ с открытым кодом...



Читайте, чтобы найти всю информацию, которая вам понадобится для использования этого DVD! Если вы новичок в Linux, откройте index.html на диске (Сторона 1) и перейдите в раздел Справка: там вы найдете мини-учебник, дающий ответы на вопросы справа.

Майк Сондерс, редактор диска
mike.saunders@futurenet.com

- » Что такое Linux?
- » Что такое дистрибутив?
- » Загрузка ПК с DVD
- » Разбиение жесткого диска на разделы
- » Навигация по файловой системе
- » Использование командной строки
- » Установка программ

Web-браузер

Firefox 3.6

Браузер *Firefox* несколько лет лидировал в сообществе свободного ПО, и вполне заслуженно: это отличная программа. Однако симпатии пользователей Linux начали перетягивать *Chrome* и *Opera*, благодаря своим головокружительно быстрым движкам JavaScript, и команде *Firefox* пришлось поднапрячься. Как следствие, в каждом релизе серии 3.x *Firefox* все более ускорялся, и 3.6 – не исключение: разработчики заявляют, что он теперь на 20 % быстрее.

Среди новый функций этого релиза – следующие:

» **Personas** – система лёгких тем, позволяющая изменять «макияж» браузера одним щелчком.

» **Программа обновления модулей расширения** – она ищет устаревшие модули (т. е. такие, которые могут некорректно работать с новым релизом), и предупреждает вас о них.

» **Полноэкранное видео HTML 5**, так что вы сможете смотреть свои потоки Ogg Theora безо всякой ерунды по краям.

Удобно, что *Firefox* – одна из простейших в установке программ: здесь нет пакетов, репозиториев и прочих сложностей. Просто скопируйте **firefox-3.6.tar.bz2** из раздела **Интернет/Firefox** на DVD в домашнюю директорию, распакуйте его и запустите **firefox** в полученном каталоге, примерно так:

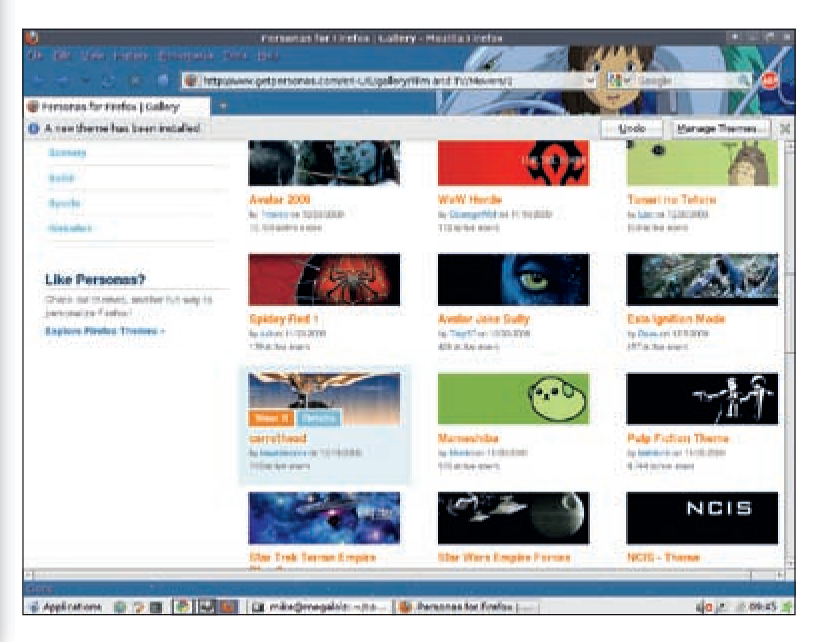
```
tar xfvj firefox-3.6.tar.bz2
cd firefox
./firefox
```

Затем можете создать ярлык для *Firefox* в вашем оконном менеджере или на рабочем столе.

Менеджер уведомлений syslog-notify

Если вам случалось заглянуть в **/var/log/messages**, вы, вероятно, видели различные предупреждения и системные сообщения об ошибках, о которых обычно вы даже не подозреваете: неудавшиеся попытки входа извне через SSH, предупреждения о неисправных носителях и т. п. В основном они безобидны, поэтому стандартный настольный Linux не делает из этого шума.

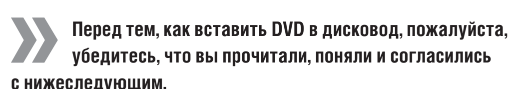
Однако иногда бывает нужно следить за компьютером более пристально, отыскивая в нем подозрительную деятель-



› *Firefox 3.6* предоставляет выбор темы одним щелчком — а тем сотни.



Очень ВАЖНО!



» Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

Диски **Linux Format** DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция **Linux Format** не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензию.

Бракованные диски

В маловероятном случае обнаружения бракованного диска Linux Format, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.

sshd[29727]
Invalid user webmaster from 173.45.108.230

› *syslog-notify*: **теперь-то вы не упустите сообщений системы!**

ность или особые предупреждения. *Syslog-notifier* добавит на ваш рабочий стол всплывающий «пузырь», где отображаются сообщения, отправляемые в системный журнал. Вы можете потребовать выводить все сообщения (что, впрочем, скоро станет раздражать вас) или только критические. Настройка делается по спецификациям freedesktop.org, так что оно весьма органично впишется в ваш менеджер окон или рабочий стол. Исходный код вы найдете в разделе **Система** на DVD.

Борьба с СЗК
GMouseTool

Пользователи компьютера подвержены синдрому запястного канала (СЗК), и одним из главных обидчиков является мышь. С точки зрения эргономики повлиять на грызуна вряд ли удастся – как форму ни меняй, все равно приходится гнать этот комок пластика на несколько сантиметров за раз, сотни раз на дню, а каждый щелчок – удар по нервам и сухожилиям, передающийся вверх по руке.

GMouseTool (раздел **Рабочий стол**) занимается только одним аспектом проблемы: нажатием на кнопки манипулятора. Он отслеживает движения мыши, и стоит вам ее остановить, как автоматически генерируется нажатие на кнопку. Итак, наведите мышь на нужную кнопку, остановите мышь, и через секунду кнопка будет нажата. Эта система работает на удивление хорошо, но все же рекомендуем оставлять курсор на свободном месте экрана (в идеале – в углу окна или панели), не то ложные срабатывания помешают вашей работе.

Помимо одиночных щелчков, можно заставить *GMouseTool* генерировать клики правой кнопкой и даже использовать его для перетаскивания. Если вы стали ощущать онемение или боль от щелканья мышью, применение этого инструмента всего пару часов в день может избавить вас от больших проблем в будущем. Он должен работать в любой среде рабочего стола и любом менеджере окон, но у пользователей KDE уже, возможно, установлен подобный инструмент – *KMouseTool*.

Рабочая среда
Исходник KDE 4.4

Кстати о KDE: уже доступен очередной релиз, переименованный в «Software Compilation». 4.4 включает новый интерфейс для нетбуков, интеграцию с социальными сетями, 7293 исправления ошибок, 1433 реализованных по запросам функции. KDE 4.4 – проект крупный и сложный, и установить его – не шуточное дело: большинству пользователей придется ждать появления обновленных пакетов для их дистрибутивов. Ну, а если вы уверенно компилируете коды с нуля, возьмите исходники для основных компонентов в разделе **Рабочий стол/KDE** на DVD.

Редактор изображений
MyPaint 0.8

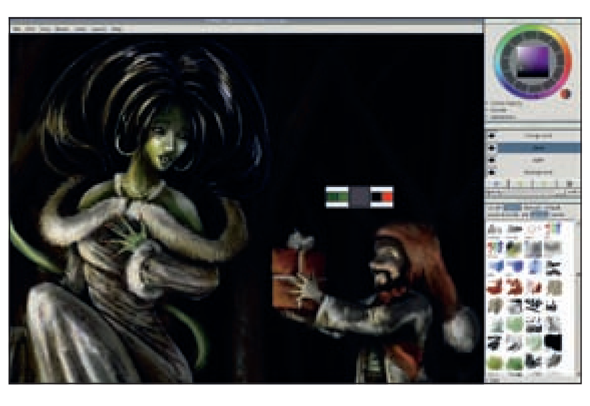
Мы всегда рады видеть альтернативу *GIMP*, а *MyPaint* обещает многое. Он умеет работать с сенсорными планшетами, и можно даже создавать кисти, реагирующие на скорость или силу нажатия. Холст здесь не ограничен размером, и его не придется масштабировать.

Для установки *MyPaint*, скопируйте файл **.tar.bz2** с **LXF**DVD в домашнюю директорию и распакуйте его. Внутри находится текстовый файл **README**, поясняющий команды, вводимые для компиляции исходного кода. Сообщения об ошибках уведомят вас о пропуске чего-то необходимого программе – тогда можно найти нужную зависимость в своем менеджере пакетов и предпринять повторную попытку. Раздел **Справка/Новичку в Linux** на **LXF**DVD предлагает дружелюбные к новым пользователям руководства по работе в командной строке и установке программ, так что при возникновении сложностей загляните туда.

Язык программирования
NewLisp 10.1.11

Небольшой вопрос: нравится ли вам следующий код?

```
(defmethod maybe-move-to-next-song
(song (playlist playlist))
(with-playlist-locked (playlist)
(when (still-current-p song playlist)
(unless (at-end-p playlist)
...
```



› **Любите порисовать, да тошнит от интерфейса *GIMP*? Возможно, *MyPaint* – приложение вашей мечты.**

Если ответ – «да», есть шансы, что вы фанат Lisp, и для вас этот язык программирования не нуждается в представлении. А если вы недоумеваете, не нарисовано ли такое изобилие скобок чисто для хохмы, знайте: Lisp – один из старейших языков программирования (он появился в 1958 г.). Ныне Lisp не столь широко распространен, как C++, Java и компания, но его связные списки, а также взаимозаменяемость кода и данных приобрели ему много сторонников. Например, на Lisp написана львиная доля текстового редактора *Emacs*.

NewLisp разработан, чтобы «сохранить все волшебство традиционного Lisp, но упростить освоение и использование». На **LXF**DVD помещен исходный код, а также Deb-пакет для Debian, Ubuntu и прочих систем на базе Debian и на редкость дотошное руководство по интерпретатору **newlisp_manual.html**.

Документация
Архив Ответов

И, наконец, обращаем ваше внимание на нашу постоянную рубрику в разделе **Справка**. Загляните в архив Ответов – там найдутся ответы и решения для огромного числа проблем и вопросов, взятые из журнала Linux Format. Если вы столкнулись с проблемой в Linux, откройте страницу и выполните поиск: возможно, там найдутся для вас уже готовые ответы и решения.

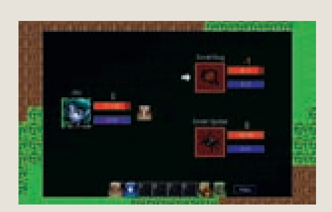
И это еще не все!
Пять классных игр

В Linux не так много коммерческих игр, но они все же существуют. Возьмем, например, гонки на мотоциклах *Mad Skills Motocross*. Мы включили демо-версию в раздел **Игры**. Чтобы сыграть в нее, скопируйте файл **.tar.gz** в домашнюю директорию, распакуйте его, перейдите в получившийся каталог и запустите **msm.sh**. Надоест – просто удалите директорию.

Далее идет *Jag*, разноцветная головоломка в стиле *KDiamonds*. Перед вами панель с разными изображениями, и вы должны передвигать картинки, составляя пары, которые будут исчезать.

А вот *Spacejunk* усадит вас в кресло пилота космического корабля, и вы будете витать в гравитационном поле планеты, собирая мусор. И если вы мечтали улечься на криогенное сохранение, чтобы году этак в 4000 получить работу мусорщика, потренируйтесь!

Еще у нас есть *Retrobattle* – подражание классическим играм *Ice Climber* и *Clu Clu Land* от NES, а также *Choria* – программа, ничуть не скрывающая свою механику ролевой игры онлайн. Вы выполняете всякую работу, убиваете монстров, потом опять работаете. Уф-ф.



› *Choria*: **«Наконец-то целью ролевой игры стал труд».**

На диске

Больше дистрибутивов, чем вы способны перелопатить

Дистрибутивы Linux

Фейерверк миниатюр

Скорее всего, в нашем Сравнении на стр. 16 вы уже видели восемь супербыстрых, не жадных до памяти дистрибутивов Linux. Ура – все они есть на DVD, и особый бонус – пять из них загружаются прямо с диска! Мы объяснили *Isolinux* и *Grub*, кто здесь главный, и позаботились, чтобы с DVD загружалось нечеловечески много дистрибутивов.

Раз дистрибутивы рассматривались в Сравнении, мы здесь не будем вдаваться в подробности насчет их функций – просто поясним, как заставить их работать с DVD. Итак, начнем…

» Lubuntu

Эта основанная на *LXDE* производная Ubuntu работает в режиме Live непосредственно с **LXF**DVD, так что вы сможете познакомиться с ней, не устанавливая на жесткий диск, как и другие вариации Ubuntu. Запустите компьютер, установив DVD в привод, и нажмите на Enter в загрузочном меню. Если последнее не появляется, вам, возможно, придется перейти в настройки BIOS и изменить порядок загрузки – в большинстве случаев надо нажать F2, Esc или Del, когда появится экран BIOS во время старта компьютера.

Попав в рабочий стол Lubuntu, ознакомьтесь с набором имеющихся программ, а если решите установить Lubuntu на жесткий диск, дважды щелкните мышью по значку Install на рабочем столе и следуйте подсказкам. После установки, при перезагрузке, когда появится окно входа в систему, выберите «Lubuntu» в выпадающем меню «Desktop» [Рабочий стол] внизу слева. (В противном случае вам будет сообщено, что рабочего стола нет). Это понадобится сделать только один раз.

› LXFDVD этого месяца — сущий монстр с пятикратной загрузкой: пять дистрибутивов Linux загружаются в режиме Live прямо с диска.

» Slitaz

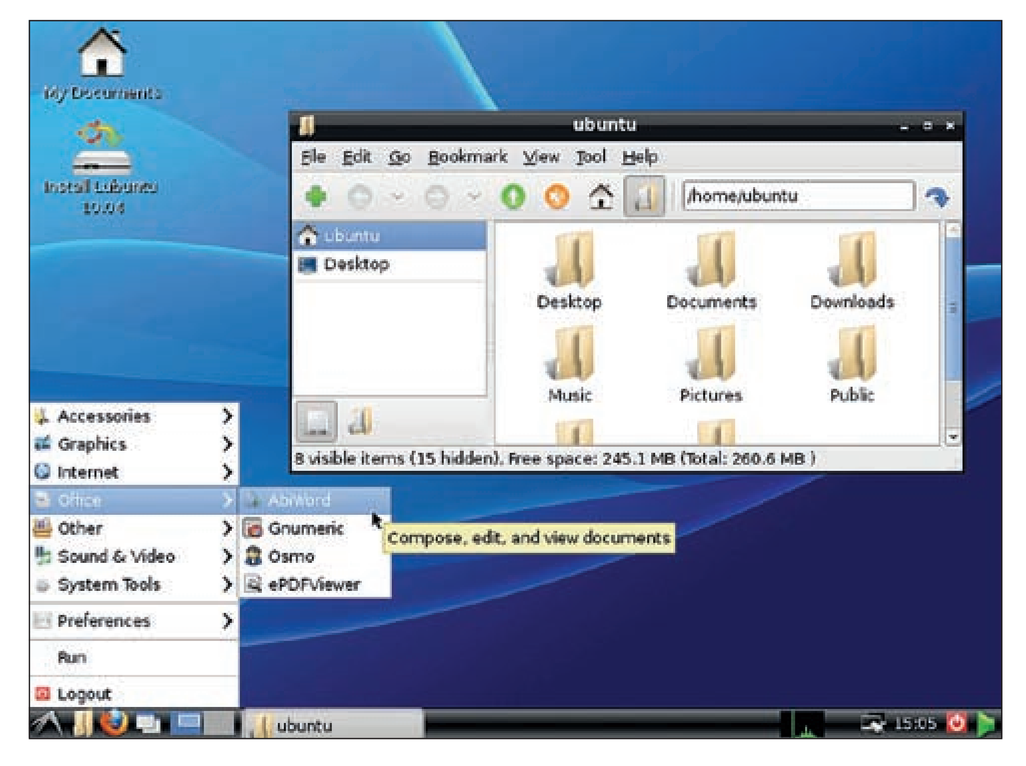
Это – второй дистрибутив, загружаемый с DVD. Выбрав его из меню, вы должны будете назвать язык и раскладку клавиатуры. После этого Slitaz предложит выбрать разрешение экрана; по умолчанию, оно довольно велико – 1600×1200, и если у вас старая машина, вы, возможно, захотите взять нечто поменьше. Третье число (24, 16 и т.д.) – глубина цвета; если у вас мало видеопамяти, опять же берите поменьше.

На рабочем столе можно щелкнуть по кнопке Menu внизу слева, чтобы познакомиться с включенными программами. Решив установить Slitaz на жесткий диск, перейдите в Menu > System Tools > Slitaz Installer [Меню > Система > Установщик Slitaz]. Этот инсталлятор работает в текстовом режиме и требует некоторых знаний о Linux: например, он попросит вас явным образом указать тот раздел, куда вы хотите установить дистрибутив. Для навигации по интерфейсу программы установки используйте клавиши управления курсором и табуляцию.

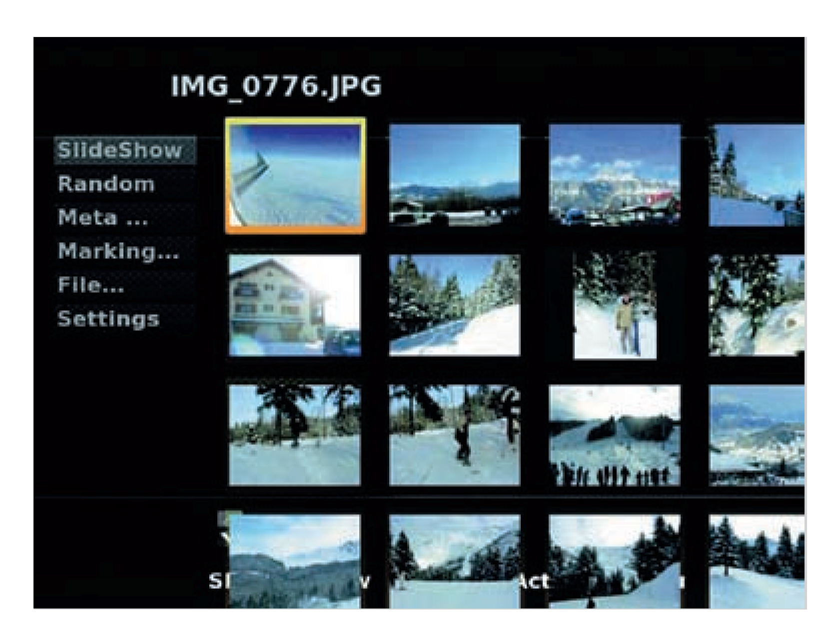
» Puppy Linux

Этот жизнерадостный дистрибутивчик поставляется на **LXF**DVD в двух форматах: можно загрузить его прямо с диска и познакомиться с ним, а если захотите установить его на винчестер или работать с ним на старой машине, на которой есть только привод CD-ROM, можете взять ISO-образ из раздела **Дистрибутивы/Puppy** на DVD и записать его на CD-R. Ваш любимый инструмент для записи дисков должен иметь опцию прожига ISO-образов.

Загрузив Puppy, вы получите возможность выбрать один из двух *X*-серверов



› Lubuntu взял за основу Ubuntu (то есть Debian) и добавил легковесный рабочий стол и приложения.



› Mythbuntu: всего за пару щелчков старый ПК из чулана превратится в мощный поющий и танцующий мультимедийный центр.

в Video Wizard [Видеомастер]. Если вы знакомы с *X*, прочитайте сопроводительный текст, чтобы указать нужный, а если нет, то просто выберите опцию «Xvesa». При этом Puppy Linux воспользуется базовым видеодрайвером, который чуть медленнее, чем специализированный, зато работает практически со всеми мониторами. Для установки Puppy запишите CD-R версию, загрузитесь с нее и откройте универсальный установщик: Menu > Setup > Puppy [Меню > Настройка > Puppy].

» Damn Small Linux

Он прост, проще некуда: выберите его из меню загрузки на DVD – и вы уже на рабочем столе. Внизу слева – меню для запуска включенных в него программ, а установить его можно через Menu > Apps > Tools > Install to Hard Drive [Меню > Приложения > Инструменты > Установить на жесткий диск]. Если хотите инсталлировать дистрибутив на старую машину без DVD-привода, возьмите ISO-образ в разделе **Дистрибутивы/Damn Small** на DVD: вы сможете записать его на CD-R и загрузиться с последнего.

» Unity Linux

В загрузочном меню **LXF**DVD есть две опции для Unity Linux, а прокрутив ниже, вы найдете еще. В случае проблем со стартом обычного «Unity Linux» попробуйте варианты безопасного режима для видео и оборудования. Если у вас 1 ГБ ОЗУ, попробуйте опцию Copy To RAM, которая перенесет весь дистрибутив в память вашей машины, и он будет работать с невероятной скоростью.

На экране приглашения Unity вы можете зайти в качестве гостя с именем и паролем «guest» (напоминание об этом имеется в верхнем левом углу). Захотите войти как администратор – тогда пароль «root», а значок для программы установки – на рабочем столе.

» CrunchBang, Tiny Core, Vector

И, наконец, еще три дистрибутива записаны на DVD в формате ISO для прожига на CD-R и последующей загрузки. Помните, что просто скопировать файлы ISO на CD нельзя: нужно прожечь их именно как ISO-образы (т.е. перенести каждый байт). Более подробную информацию о том, как это делается, ищите в документации своей программы для записи дисков. В командной строке (от имени root):

```
cdrecord -v -dao dev=/dev/cdrom
filename.iso
```

Дистрибутив для мультимедийного центра

Mythbuntu 9.10

Хотите превратить списанный ПК в мощную домашнюю развлекательную систему? MythTV и Linux – наилучшая комбинация для этого, но проблема в одном: как отдельная программа, MythTV знаменит сложностью настройки. К счастью, вам не нужна Оксбриджская [Oxbridge – ironич. гибрид «Оксфорд» и «Кембридж», названий двух самых знаменитых университетов Великобритании, – *прим. пер.*] степень по MythTV'ологии, чтобы все-таки заставить его работать, а все благодаря Mythbuntu, союзу вечно популярного дистрибутива Ubuntu и вышеупомянутого пакета мультимедиа.

На **LXF**DVD, в разделе **Дистрибутивы/Mythbuntu**, вы найдете ISO-образ релиза 9.10. Можете записать его на CD-R и загрузиться, как описано в предыдущем разделе. Добравшись до рабочего стола, поэкспериментируйте с MythTV, дважды щелкнув по значку Live CD Frontend, а установить его на жесткий диск поможет значок Install.

Если мир MythTV вам неведом, у нас есть всестороннее онлайн-руководство на http://bit.ly/cSUUnl – оно объяснит вам, как установить Mythbuntu и настроить MythTV для потокового вещания, записи ТВ, просмотра фото, игр и многого другого. LXF

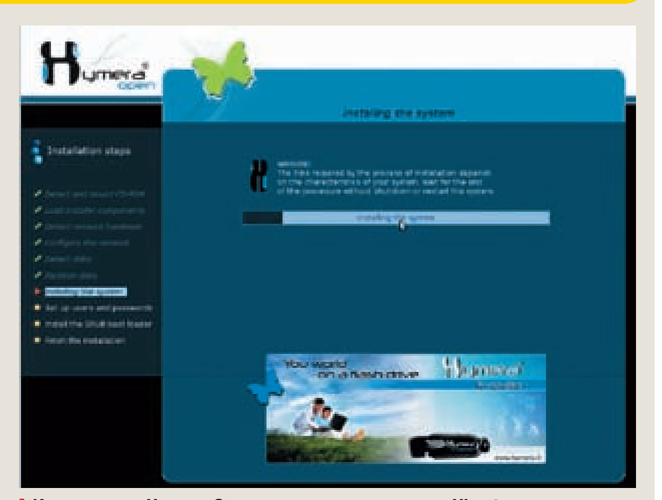
Вторая сторона

Hymera Open

Про итальянский дистрибутив Hymera немало сказано в обзоре на стр. 11. Поэтому здесь мы не будем расписывать его достоинства, а пройдемся по основным этапам установки.

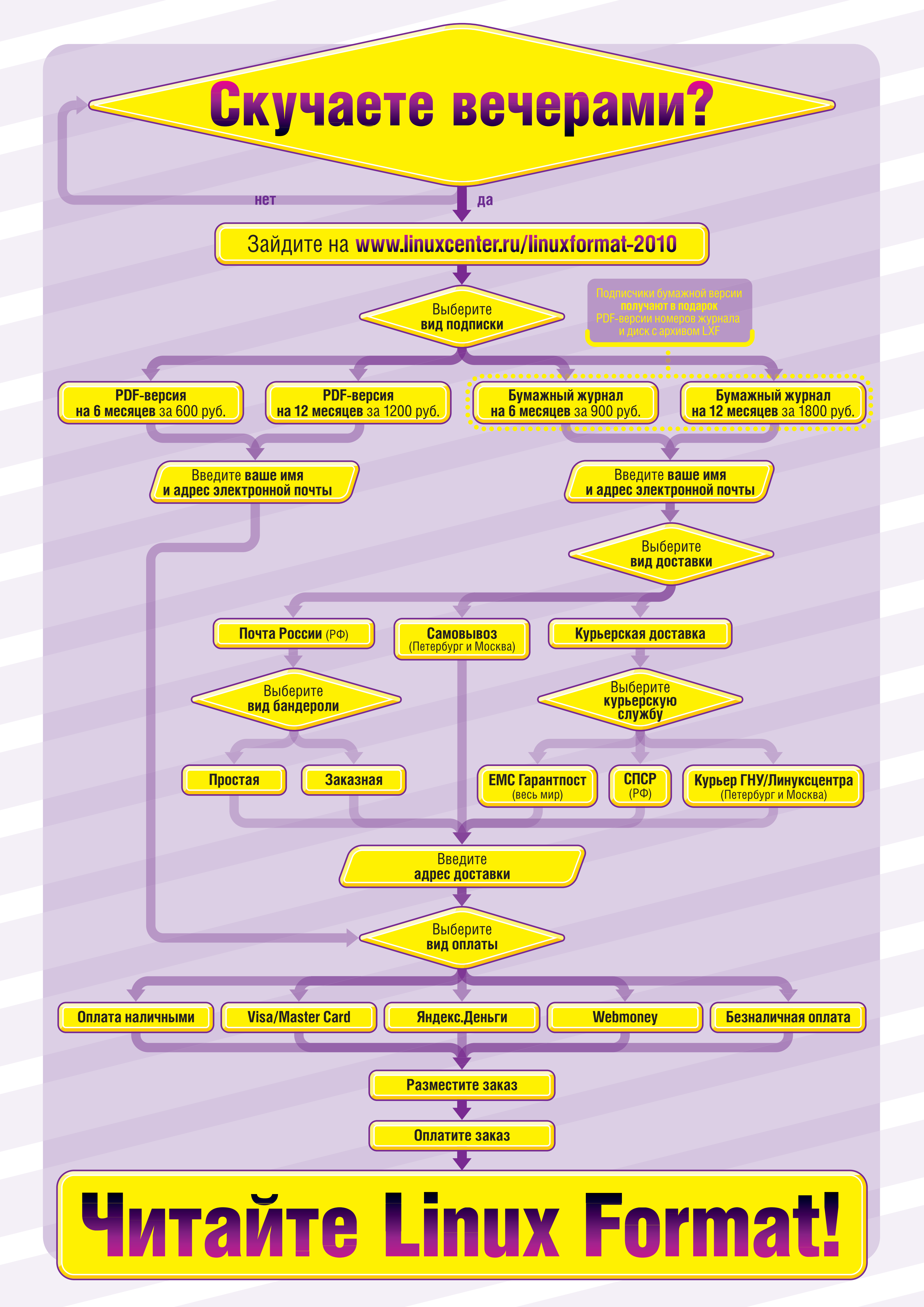
Загрузите компьютер со второй стороны **LXF**DVD и примите лицензионное соглашение, ничем не ограничивающее обычного пользователя. Далее, введите сетевое имя, домен и разбейте жесткий диск на разделы. В автоматическом (guided) режиме используется весь диск и создается единственный корневой раздел и раздел подкачки. В ручном (manual) предоставляется полный контроль за разбивкой с возможностью выбора типа файловой системы и точек монтирования разделов (выделите для / не менее 8 ГБ). Дождитесь, пока инсталлятор отформатирует разделы и скопирует необходимые файлы. Введите пароль суперпользователя и создайте обычного пользователя для повседневных нужд. Наконец, установите загрузчик и перезагрузитесь.

При наличии подключения к сети можно сразу же изменить язык системы на русский с помощью утилиты Languages Manager [Менеджер языков], находящейся в подменю System Tools [Системные инструменты]. В списке слева нужно дважды щелкнуть на русском и пометить его как язык по умолчанию. После перезагрузки Hymera будет русифицирована. Поддержка русского языка пока неполная – многие программы и ярлыки остаются англоязычными.



› Инсталлятор Hymera Open не сложнее такового в Ubuntu, да и выглядит симпатично.

Если вы планируете опробовать Hymera Open в VirtualBox, то можете столкнутся с неработающим драйвером видеокарты. Самый простой выход – скачать готовый образ виртуального диска с предустановленной Hymera Open: http://www.mininova.org/tor/2989776.


Скучаете вечерами?
нет
да
Зайдите на www.linuxcenter.ru/linuxformat-2010
Выберите вид подписки
Подписчики бумажной версии получают в подарок PDF-версии номеров журнала и диск с архивом LXF
PDF-версия на 6 месяцев за 600 руб.
PDF-версия на 12 месяцев за 1200 руб.
Бумажный журнал на 6 месяцев за 900 руб.
Бумажный журнал на 12 месяцев за 1800 руб.
Введите ваше имя и адрес электронной почты
Введите ваше имя и адрес электронной почты
Выберите вид доставки
Почта России (РФ)
Самовывоз (Петербург и Москва)
Курьерская доставка
Выберите вид бандероли
Выберите курьерскую службу
Простая
Заказная
EMC Гарантпост (весь мир)
СПСР (РФ)
Курьер ГНУ/Линуксцентра (Петербург и Москва)
Введите адрес доставки
Выберите вид оплаты
Оплата наличными
Visa/Master Card
Яндекс.Деньги
Webmoney
Безналичная оплата
Разместите заказ
Оплатите заказ
Читайте Linux Format!

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Внимательно прочтите это перед тем, как использовать LXFDVD!

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

» **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;

» **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;

» **имя_программы-1.0.1.tar.gz** – обычно это исходный код;

» **имя_программы-1.0.1.tgz** – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;

» **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;

» **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

» **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

» **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

» **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается…

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Апрель 2010 LXFDVD 130

LINUX FORMAT

Hymera Open и Pardus 2009.1

Эти региональные дистрибутивы уже готовы приобрести международную значимость

» Рабочие столы KDE (Pardus) и Gnome (Hymera) в нескольких вариантах

» Оригинальный подход к организации и оформлению рабочего пространства (Hymera)

» Собственные средства настройки и менеджер пакетов (Pardus)

А ТАКЖЕ: фирменные драйверы, мультимедиа-кодеки и поддержка Flash «из коробки»!

Апрель 2010 LXFDVD 130

LINUX FORMAT



» **Mythbuntu 9.10** Создайте домашний центр развлечений сегодня

» **Mad Skills Motocross** Гонки на мотоциклах (демо-версия)

» **Ответы** Архив рубрики LXF: проблемы Linux решены!

А ТАКЖЕ: средства разработки, настольное ПО и еще игры!

DVD-приложение к журналу Linux Format №4 (130). Апрель 2010

Содержание

LINUX FORMAT

Сторона 1

Сторона 2

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИИ? Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь пожалуйста по адресу disks@linuxformat.ru

Тираж изготовлен ООО «Уральский электронный завод». 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАФ 77-15.



Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика.



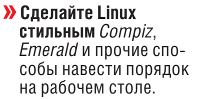
Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам, но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в Линуксцентр по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска **LXF** может пройти не более нескольких минут!

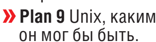
Прямо сейчас для заказа доступны следующие номера:



LXF126/127
Январь 2010



» **Сделайте Linux стильным** *Compiz*, *Emerald* и прочие способы навести порядок на рабочем столе.



» **Plan 9** Unix, каким он мог бы быть.

» **Подводные лодки** Linux в исследовательских аппаратах.

» **Wireshark** Прослушивайте трафик и защищайтесь от перехвата.

LXFDVD: Ubuntu 9.10, Mandriva 2010 Free и OpenSUSE 11.2

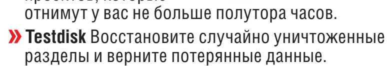
Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_126_127/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_126_127/

LXF128
Февраль 2010



» **Курс молодого бойца** Самое подробное введение в Linux для начинающих.



» **Десятиминутки** Девять проектов, которые отнимут у вас не больше полутора часов.

» **Testdisk** Восстановите случайно уничтоженные разделы и верните потерянные данные.

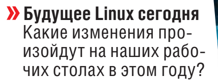
» **Clutter** Создавайте приложения для платформы Moblin или для Gnome 3.0.

LXFDVD: Fedora 12 и подшивка **LXF** за первую половину 2009 года!

Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_128/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_128/

LXF129
Март 2010



» **Будущее Linux сегодня** Какие изменения произойдут на наших рабочих столах в этом году?

» **Ядро ждет помощи от ВАС** Впишите свое имя в списки разработчиков Linux.

» **Android** Создание приложений для мобильной платформы Google.

» **gPhoto и PTP** Управляйте цифровой камерой с компьютера.

LXFDVD: Linux Mint 8, FreeBSD 8.0, Calculate Linux 10.2, а также книги в PDF

Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_129/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_129/

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала – оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru получают электронную версию в подарок!

Спешите на www.linuxformat.ru/subscribe!





По чему учить Linux?

Александр Казанцев сделает обзор существующих учебных пособий и оценит их «пригодность» для проведения уроков на базе свободного ПО.



Наш эксперт

Александр Казанцев
К.т.н., доцент кафедры Информатики Глазовского государственного пединститута, руководитель проекта EduMandriva, автор локализаций (и разработок) для Mandriva, *Klavaro*, *Gambas2* и *LXDE*.

Занятия в наших учебных заведениях ведутся не по прихоти преподавателя (хотя, при должном соблюдении всех требований, никто не запрещает вести «авторские курсы»), а регламентируются соответствующими законами и подзаконными актами.

В школе, двумя изначальными составляющими проведения уроков будут примерная программа по информатике, стандарт по информатике и рекомендованный (или допущенный) учебник, по которым будут вестись занятия и строиться их содержание. Так как учебники на новый учебный год назначаются по поговорке «готовь сани летом», то федеральный перечень рекомендованных или допущенных учебников на новый 2010/11 год был утвержден Приказом № 822 от 23 октября 2009 года. С учетом того, что учебное заведение может остановить свой выбор как на свободном, так и на проприетарном программном обеспечении, применимость учебников и сопутствующих методик будет не последним фактором в принятии решения об их использовании.

Давайте проведем анализ рекомендованных учебников, применимых для проведения занятий по информатике и ИКТ в основном общем и полном среднем образовании. Для вынесения решения мы рассмотрим учебники по следующим критериям: ОС и дистрибутивонезависимость, возможность использования учебника с СПО-решениями, отсутствие привязки методик или программных средств к операционной системе, наличие примеров с использованием СПО. Анализ будем проводить по содержательной части учебников для 10-11 класса, при необходимости делая «отсылку» на 8-9 классы, так как канва повествования и практических занятий у одних и тех же авторов в большинстве случаев будет совпадать. Мы не будем касаться соответствия учебникам стандартам или их применимости для подготовки к ЕГЭ: наша основная задача, бросив беглый взгляд, понять: а возможно ли использовать эти учебники при установке на школьные и домашние машины учащихся только СПО-решений, а в идеале – только Linux.

Наш выбор

Итак, прочитав указанный выше приказ и приняв наши допущения, мы получаем следующих претендентов на звание «СПО-учебник по информатике»:

» Гейн А. Г., Ливчак А. Б., Сенокосов А. И. и др. Информатика и ИКТ, издательство Просвещение, 10-11 классы
» Макарова Н. В., Николайчук Г. С., Титова Ю. Ф. / Под ред. Макаровой Н. В. Информатика и ИКТ, издательство Питер Пресс, 10-11 классы
» Семакин И. Г., Хеннер Е. К. Информатика и ИКТ, издательство БИНОМ, 10-11 классы
» Угринович Н. Д. Информатика и ИКТ, издательство БИНОМ, 10-11 классы

В данной статье мы не рассматриваем учебники для более младших классов, так как это тема для отдельного материала, требующая анализа понятия учебных планов и программ нового поколения. Также для упрощения мы объединим учебники авторов за 10-11 классы в части базовых и профильных курсов и будем рассматривать их в рамках единого УМК; и, при необходимости, обратим внимание на дополнительные книги комплекта, такие как практикумы или книги учителя.

Если внимательно изучить примерную программу (смотрите врезку «*Стандарт и СПО*»), то она (как и остальные нормативные документы) не содержит требований к определенному про-



Стандарт и СПО

Как мы уже говорили, примерная программа не несет привязки к конкретному ПО, но может вызвать трудности при адаптации ее к использованию со свободными программными продуктами. Ниже приведен кусок из «адаптации» примерной программы (с рекомендациями) для использования с Linux-решениями, которая была выполнена в рамках проекта EduMandriva (www.edumandriva.ru) и ориентирована на этот дистрибутив. Курсивом даны комментарии, поясняющие применение СПО. Полностью примерную программу можно посмотреть здесь: http://wiki.edumandriva.ru/wiki/index.php/Программа_по_Информатике_и_ИКТ_для_10-11_классов

Средства ИКТ и их применение (48 часов)

» Архитектуры компьютеров и компьютерных сетей (3 часа)
Программная и аппаратная организация компьютеров, других средств ИКТ и их систем.

Сделать упор на понятия открытых и закрытых архитектур.

» Понятия открытого, свободного и проприетарного ПО. Юридические и моральные аспекты использования свободного и открытого ПО. (+4 часа из резерва)

Виды программного обеспечения. Авторское право и патенты. Проблемы авторского права в мире программного обеспечения. Ответственность за нарушение авторских прав. Понятия свободного и открытого ПО. Достоинства и недостатки использования свободного/открытого ПО. Движения GNU и Open Source.

» Операционные системы (5 часов)
Функции операционной системы. Основные виды и особенности операционных систем. Понятие о системном администрировании.

Рассказать об основных операционных системах (Windows/Mac OS/Unix/Linux), но упор сделать на Linux. Рассказать о преимуществах администрирования Linux-систем и существующих программах, его упрощающих (Webmin, Mandriva DirectoryServer).

» Практика программирования (16 часов)
Язык программирования. Понятность программы. Внесение изменений в программу. Структурное программирование, объектно-ориентированный подход. Ошибки, отладка, построение правильно работающих и эффективных программ. Этапы разработки программы.

За основу взять один из языков: Basic (Gambas2), Pascal (FreePascal, Lazarus), Python или C/C++.

» Практика применения ИКТ (6 часов)
Планирование и проектирование применения ИКТ; основные этапы, схемы взаимодействия. Типичные неисправности и трудности в использовании ИКТ. Профилактика оборудования. Комплектация рабочего места средствами ИКТ в соответствии с целями его использования. Оценка числовых параметров информационных объектов и процессов, характерных для различных областей деятельности.

Сделать упор на проблемы использования Linux-систем и развертывания компьютерных классов на Linux. Указать на проблемы с несовместимостью аппаратно-программных средств.

» Организация и поиск информации (6 часов)
Представление о системах управления базами данных, поисковых системах в компьютерных сетях, библиотечных информационных системах. Компьютерные архивы информации: электронные каталоги, базы данных. Организация баз данных. Примеры баз данных: юридические, библиотечные, здравоохранения, налоговые, социальные, кадровые. Использование инструментов системы управления базами данных для формирования примера базы данных учащихся в школе.

*Привести примеры **открытых** баз данных (MySQL, PostgreSQL), а также инструментов. Дать ссылки на открытые архивы информации (например, сайты проекта Wikipedia (http://ru.wikipedia.org), библиотеки Линуксцентра (http://www.linuxcenter.ru/lib/), проекта «Виртуальная энциклопедия «Линукс по-русски» (http://rus-linux.net/). Использовать для примера создание базы школы в OpenOffice.org Base или MySQL в связке любым языком программирования.*

Практикум (140 часов + 6 часов из резерва)

» Технологический проект: установка, сопровождение, техническое обслуживание средств ИКТ.
В проекте силами учащихся под руководством взрослых может осуществляться работа в сфере ИКТ, требующая базовых технических знаний и умения понимать технические инструкции

Установка и настройка EduMandriva Linux. Настройка пользователей и базовых сервисов в EduMandriva Linux. Установка и удаление программ. Базовое обслуживание системы.

граммному обеспечению. Единый государственный экзамен по информатике также не несет четких отсылок к конкретным программным продуктам (напротив, дается достаточная свобода по использованию сред программирования, и все задания составлены инвариантно к программным продуктам – см врезку: «*ЕГЭ – есть ли проблемы с Linux?*»). При изложении материала учебники также, по сути, не должны содержать привязки к конкретному ПО, но в части практических примеров авторы могут иметь какие-то предпочтения, в основном продиктованные федеральными программами в рамках национального проекта «Образование – Первая Помощь» и «Пакет свободного программного обеспечения».

Абстракционизм

Итак, приступим. Первым к нам на «растерзание» поступает учебник **Гейн А. Г., Ливчак А. Б., Сенокосов А. И. и др. «Информатика и ИКТ»**, выпущенный издательством Просвещение.

Авторы данного учебника, с одной стороны, решили пойти по изначально заложенным в стандарте принципам и дистанцироваться от программного обеспечения и даже конкретных реализаций языков программирования. В результате изложение в большей части ведется на абстрактных примерах вида Текстовый редактор, Электронные таблицы и т. п. С другой стороны, практическая реализация примеров целиком выполнена с использованием проприетарных продуктов в операционной системе MS Windows. Плохо это или нет? С одной стороны, учебник может «жить» без изменений достаточное время, так как базовые принципы прикладных программ остаются постоянными с начала 90-х годов (а некоторые вещи вообще придуманы в 50–60-е годы XX века). Кроме того, абстрагирование от конкретных программных продуктов позволяет использовать этот учебник совместно с Linux или СПО-решениями. С другой стороны, это требует от преподавателя самостоятельной разработки практической составляющей.

В части программирования используется собственная разработка – исполнитель Паркетчик, который хотя и написан под Windows, но прекрасно чувствует себя в *Wine* (к слову, автор данной статьи пытался получить исходные коды Паркетчика для адаптации под Linux, но, похоже, они или были утеряны, или авторы учебника не заинтересованы в этом).



Совместимо с Windows

Перейдем к следующей группе: **Макарова Н. В., Николайчук Г. С., Титова Ю. Ф. / Под ред. Макаровой Н. В. «Информатика и ИКТ»** (базовый уровень), издательство Питер Пресс. Учебники этих авторов имеют целевую проприетарную направленность и построены на использовании продуктов Microsoft, что явно видно из изучения их содержания. Целевыми системами, на которые ориентированы практические задания, являются составляющие Microsoft Office (*Word*, *Access*, *Excel* и *PowerPoint*) в версии 2003 (что очень важно для нас), а также Visual Basic как язык и среда программирования.

С другой стороны, как это ни парадоксально, «нацеливание» учебников на данные продукты может сыграть и положительную роль. Как мы писали ранее в наших статьях (см. «Школу **LXF**» за 2009 год), *OpenOffice.org* с помощью расширений можно «подогнать» к возможностям *MS Office*, а работа с интерфейсом 2003 версии (в отличие от «лент» 2007–2010 версий *MS Office*) будет похожей, вплоть до совпадения пунктов. Графика в учебниках 10 класса тоже изучается на примере *Word*, и использование *Draw* даже подчеркнет преимущества свободного ПО.

Больше проблем вызовет практикум по программированию и использование в качестве основы изучения баз данных СУБД *Access*, но в данном контексте грамотный преподаватель сможет адаптироваться к свободным продуктам, используя *Gambas* и *Base*. Следует учесть и то, что в данном случае изучение операционных систем на базовом уровне не выполняется, и различий, на какой платформе использовать прикладное ПО, не имеется (чего нельзя сказать про учебники для 8-9 классов, имеющих 100 % Windows-направленность в части рассмотрения операционных систем).



Сделай сам

Наш следующий «конкурсант» – **Семакин И. Г., Хеннер Е. К. «Информатика и ИКТ»** (базовый уровень) издательства БИНОМ. Данный учебник оставляет двойственное впечатление. С одной стороны, изложение материала в теоретической части не привязано к конкретным программным продуктам (идет упоминание о текстовом процессоре, табличном процессоре, программе подготовки презентаций); с другой (особенно если брать во внимание практикум) – идет привязка к офисным продуктам Microsoft, о чем авторы честно предупреждают, но также говорят о возможности адаптации заданий к Linux и открытому ПО. Ситуация с базами данных и СУБД повторяет описанную выше в учебниках Макаровой. Поэтому останавливаться в деталях на данном учебнике мы не будем.



Почти про Linux

И, наконец, поговорим об учебнике, о котором (так или иначе) слышали, наверное, все пользователи Linux, даже те, кто не имеет отношения к преподаванию. Я имею в виду **Угринович Н. Д. «Информатика и ИКТ»** (базовый уровень), выпущенный издательством БИНОМ.

Это единственный учебник, который имеет в составе методического пособия для учителей компакт-диск Linux CD, а также содержит информацию по работе в свободном ПО – *OpenOffice.org*, *Scribus*, *GIMP*, *Audacity*, *SeaMonkey*. В свою очередь, многие вещи рассмотрены только с позиции проприетарных продуктов, и упор делается на Windows CD, как, например, в случае Flash или CAD *Компас*. Определение параметров компьютера, практика с файловой системой, работа в векторном редакторе и CAD-системе и другие вещи, в которых свободное ПО показывает себя доста-



точно хорошо, просто обойдены вниманием авторов и потребуют самостоятельной разработки уроков преподавателем (вопрос заключается в том, зачем здесь нужен Linux CD).

Также удручает и то, что программирование рассматривается в конкретных проприетарных средах (Visual Basic и Turbo Delphi) – при наличии Gambas2 и Lazarus/FreePascal. Факт рассмотрения операционной системы Linux и свободного ПО (особенно в так называемых «новых» учебниках для 11 класса) не может не радовать, но упор в части программирования на технологии с # в названии огорчает – все примеры рассмотрены на основе реализаций C#, J# и VisualBasic, которые (судя по схемам) являются венцами эволюции языков.

Подведем итоги

Итак, победителями данного «смотра» стали учебники под авторством Угриновича Н. Д., но, с другой стороны, победа эта относительная. К сожалению, ситуация такова, что не существует ни одного учебника, который можно было бы на 100 % использовать совместно с СПО без должной переработки преподавателем, что требует от последнего достаточных знаний по Linux и другим свободным продуктам. С методическими составляющими – практикумами, книгами учителя, элективными курсами – ситуация еще более плачевная. Преподаватель, решивший использовать Linux в школе, вынужден будет самостоятельно адаптировать (хотя в случае выбора учебников Угриновича это будет сделать не так сложно) программу обучения под имеющиеся программные средства.

Пока у нас не будет в учебниках равноправия проприетарных и свободных решений, пока для преподавателей не будет законченных УМК по данному вопросу (что опять же не проблема образовательной программы или стандартов, которые как раз индифферентны к используемым программным средствам), внедрение и использование Linux и свободного ПО в образовательных учреждениях будет уделом энтузиастов. Поэтому новое поколение преподавателей должно приложить все усилия к изменению существующей ситуации. LXF



ЕГЭ — есть ли проблемы с Linux?

Посмотрев примерные задания по ЕГЭ (http://www1.ege.edu.ru/images/stories/ege2010/demo_2010/inform_demo_2010.pdf), предлагаемые выпускникам школ в этом году, можно отметить отсутствие привязки к конкретных программным продуктам. В заданиях рассматриваются вопросы, не привязанные к ОС или конкретным приложениям, а в случае необходимости используется терминология «абстрактных» инструментов – база данных, электронная таблица и т. д. Программы же представлены на трех языках, причем в процедурной форме – Паскале, Бейсике, Си и алгоритмическом, свободные реализации которых есть в любом «школьном» дистрибутиве Linux. Плюс к этому, выбор языка для решения заданий по написанию программ не ограничен вообще (можно, например, писать их на Erlang или на Perl, и они должны быть приняты к проверке и оценены наравне с остальными).

Поэтому опасения, что применение Linux может повлиять на результат подготовки к ЕГЭ являются надуманными – наоборот, использование свободного ПО дает больший простор для этой подготовки (я в своей практике не видел, чтобы на компьютерах с Windows, к примеру, были установлены одновременно компиляторы и интерпретаторы используемых в ЕГЭ языков).

Сравнительные характеристики

		Информатика и ИКТ. 10–11 классы	Информатика и ИКТ. Базовый уровень	Информатика и ИКТ. Базовый уровень	Информатика и ИКТ. Базовый уровень
	Авторы	Гейн А. Г., Ливчак А. Б., Сенокосов А. И. и др.	Макарова Н. В., Николайчук Г. С., Титова Ю. Ф., под ред. Макаровой Н. В.	Семакин И. Г., Хеннер Е. К.	Угринович Н. Д.
	Издательство	Просвещение	Питер Пресс	Бином	Бином
Рейтинг*	ОС и дистрибутивонезависимость	★★	★	★★	★★★★★★
	Возможность использования учебника с СПО-решениями	★★★★★★	★★★	★★★★★★	★★★★★★★★
	Отсутствие привязки методик или программных средств к операционной системе	★★	☆	★★	★★★★★★
	Наличие примеров с использованием СПО	☆	☆	☆	★ 10
	Возможность применения учебника без Windows	★★★★	★★	★★★★	★★★★★★
Итоговый рейтинг		★★★★	★★	★★★★	★★★★★★★★

* По десятибальной шкале.



Журнал зарегистрирован Федеральной службой
по надзору за соблюдением законодательства в сфере массовых коммуникаций
и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор
Валентин Синицын info@linuxformat.ru
Литературный редактор
Елена Толстякова
Переводчики
Илья Аввакумов, Александр Бикмеев, Юлия Дронова, Александр Казанцев, Ольга Кокорева, Светлана Кривошеина, Алексей Опарин, Валентин Развозжаев, Татьяна Цыганова
Редактор диска
Александр Кузьменков
Верстка, допечатная подготовка
Сергей Рогожников
Технический директор
Денис Филиппов
Генеральный директор
Павел Фролов
Учредители
Частные лица
Издатели
Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Взлет»
196644, Санкт-Петербург, Колпинский район, пос. Саперный, тел. (812) 462-85-85
Заказ

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Пол Хадсон [Paul Hudson] paul.hudson@futurenet.com
Редактор обзоров Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.com
Редактор диска Майк Сондерс [Mike Saunders] mike.saunders@futurenet.com
Художественный редактор Эфраин Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com
Литературный редактор Эндрю Грегори [Andrew Gregory] agregory@futurenet.com
Подготовка материалов Мо Авкати [Mo Awkati], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Энди Ченнел [Andy Channelle], Марко Фиоретти [Marco Fioretti], Джульетта Кемп [Juliet Kemp], Грег Кроа-Хартман [Greg Kroah-Hartman], Энди Хадсон [Andy Hudson], Дэниел Джеймс [Daniel James], Боб Мосс [Bob Moss], Маянк Шарма [Mayank Sharma], Ник Вейч [Nick Veitch], Леонид Алифанов, Евгений Балдин, Семен Есилевский, Андрей Боровский, Павел Воронцов, Алексей Джежора, Андрей Казанцев, Евгений Крестников, Сергей Петров, Андрей Прахов, Алексей Федорчук
Художественные ассистенты Стейси Блек [Stacey Black], Ник Кокс [Nick Cox]
Иллюстрации Крис Хидли [Cris Hedley], Крис Винн [Chris Winn], iStockPhoto

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel +44 01225 442244 Email: linuxformat@futurenet.com
РОССИЯ:
Санкт-Петербург (редакция):
Лиговский пр., 50, корп. 15
Тел.: +7 (812) 309-06-86
Представительство в Москве:
Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)
Тел/факс: +7 (499) 271-49-54
По вопросам сотрудничества, партнерства, оптовых закупок:
partner@linuxcenter.ru



BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



В майском номере

Как работает Linux

Нравится знать, как все устроено? Удовлетворите свое любопытство: мы покажем, как связаны друг с другом различные части Linux.

Тестируем: дистрибутивы с MythTV

Как заставить Linux работать в вашей гостиной, не потратив на его настройку полжизни?

DOSBox: практическое руководство

Ах, старые добрые деньки... Верните их, используя свободное ПО, чуть-чуть конфигурирования и знания наших экспертов!

Содержание последующих выпусков может меняться без уведомления – вдруг мы заиграемся в старые игры для DOS?

HOSTING NEXT LEVEL



HETZNER DEDICATED ROOT SERVER EQ 4

- Intel®Core™ i7-920 Quad-core с поддержкой технологии Hyper-Threading
- 8 GB DDR3 RAM
- 2 x 750 GB SATA-II HDD (Software-RAID 1)
- Операционная система Linux
- Windows Server Web Edition (700 руб. в месяц)
- Неограниченный трафик[1]
- Восстановление системы
- Установка из образов
- 100 GB пространства для резервных копий
- Без минимального контракта
- Стоимость установки 5900 рублей

1900 рублей в месяц

HETZNER DEDICATED ROOT SERVER EQ 8

- Intel®Core™ i7-920 Quad-core с поддержкой технологии Hyper-Threading
- 24 GB DDR3 RAM
- 2 x 1500 GB SATA-II HDD (Software-RAID 1)
- Операционная система Linux
- Windows Server Web Edition (700 руб. в месяц)
- Неограниченный трафик[1]
- Восстановление системы
- Установка из образов
- 100 GB пространства для резервных копий
- Без минимального контракта
- Стоимость установки 5900 рублей

3300 рублей в месяц

HETZNER DEDICATED ROOT SERVER EQ 9

- Intel®Core™ i7-975 Quad-core с поддержкой технологии Hyper-Threading
- 12 GB DDR3 RAM
- 3 x 1500 GB SATA-II HDD (Software-RAID 5)
- Операционная система Linux
- Windows Server Web Edition (700 руб. в месяц)
- Неограниченный трафик[1]
- Восстановление системы
- Установка из образов
- 100 GB пространства для резервных копий
- Без минимального контракта
- Стоимость установки 5900 рублей

3900 рублей в месяц

1 Трафик предоставляется бесплатно. При превышении порога 2000 GB/месяц скорость соединения будет ограничена 10 MBit/s. Стоимость аренды постоянного канала с пропускной способностью 100 MBit/s составляет 590 рублей за каждый дополнительный TB.

2 Как новый клиент, вы можете сэкономить 340 рублей на первом платеже за любой из рекламируемых здесь продуктов. Просто используйте код ваучера **051104** при совершении заказа. Предложение действительно до 19 мая 2010 года.

HETZNER ONLINE

Hosting Next Level (Хостинг нового уровня) означает, что компания Hetzner Online готова предоставить вам самые мощные решения для хостинга выделенных серверов из имеющихся сегодня на рынке. Наши предложения были разработаны, чтобы предоставить вам более высокую скорость и чрезвычайно стабильную сетевую инфраструктуру на базе наших собственных дата-центров в Германии. Благодаря лучшим ценам и непревзойденной поддержке, мы превосходим ожидания клиентов по всему миру.

www.hetzner.info
info@hetzner.com